uc3m | Universidad Carlos III de Madrid

Bachelor's Degree in Industrial Electronics and Automation
Academic Year 2019-2020

Bachelor Thesis

A Kalman Filter application for GNSS error correction in Intelligent Vehicles

---

Manuel Fraile Rodríguez

Tutor

David Martín Gómez

Leganés, 19 February 2020

ABSTRACT

This proposes a Kalman Filter application for solving errors in the positioning of a moving vehicle in an everyday urban environment that is also cost effective. The GNSS sensor is a NOVATEL FlexPak-G2. It is installed in a car and takes measurements for several kilometres providing a wide range of data from which the position and the velocity are used to construct the model. The velocity is taken as a reference and three models are proposed in order to solve the inconsistencies of the GNSS sensor.

The first model uses a Kalman Filter and is based on the hypothesis of a Rectilinear Uniform Motion. The second model uses a Kalman Filter and is based in the hypothesis of a Uniformly Accelerated Rectilinear Motion. The last of the three models uses an Unscented Kalman Filter and is based on the hypothesis of a Uniformly Accelerated Rectilinear Motion.

At the end of the thesis, the three models will be compared and discussed in order to conclude the one that is best for the desired solution.

Keywords: Kalman, singularity, correction, system.

ACKNOWLEDGMENT

This thesis is the culmination of a life period that ends today and started four years ago. All the knowledge, the hard work and the experiences lived, converge in this final thesis. But all this, would have been impossible without all the people that came along with me through this path. Therefore, thanks to the ones that stayed and the ones who arrived, all of you have a recognisable contribution to this thesis.

Thanks to David Martín for guiding me through these past months and always fitting meetings in such a complicated schedule.

Thanks to Ramón Izquierdo for being one of those who stayed and for giving legal counselling for this thesis.

Thanks to Ariadna Ferrer for being one of those who arrived and creating most of the illustrations that appear in this work.

Special thanks to my parents that supported me always, no matter the circumstances, and gave me all they had for me to pursue my goals.

TABLE OF CONTENTS

TABLE OF FIGURES

1 INTRODUCTION

In this thesis it is going to be presented a solution for error correction in wrong measurements made by a GNSS system that in this case is a NOVATEL receptor that provides an analysis of the velocity and position. The position errors will be rectified with a model constructed with Kalman Filters that provides a corrected trajectory. The main purpose of this technology is to provide a reliable GNSS solution for autonomous vehicle navigation that could help to empower this technology that is currently relying more in other technologies such as computer vision or LiDAR.

The thesis will start from the big picture, with an introduction to the autonomous vehicle scene and, in particular, to the navigation solutions that are currently at the state of the art. Following this, it will be provided a more theoretical and mathematical explanation on Kalman Filtering before explaining the three models that have been developed and finishing with the results and conclusions.

1.1 The Big Picture

It is pertinent to start discussing the situation in which the autonomous vehicle technology is nowadays in order to have an appropriate perspective to analyse the solution that is presented in this thesis. Although it is a very popular field of engineering, there are many points that should be clarified and refreshed firstly.

The scene is currently ruled by two companies that have two different value proposal that, however, seem to converge eventually. Both Tesla and Waymo stay at the forefront of the car manufacturing and the “autonomous taxi” sectors respectively. Elon Musk’s company, Tesla, has developed what they call “Autopilot”: an autonomous driving technology that is based on cameras, ultrasonic sensors and a radar to detect the obstacles surrounding the car that are understood with computer vision and machine learning. On the other hand, Waymo uses radars and cameras but also LiDAR sensors used for short-range detection. With these three, they build a system that with computer vision and machine learning is able to provide an autonomous taxi service in a series of cities around the USA. However, to what extent are their cars autonomous?

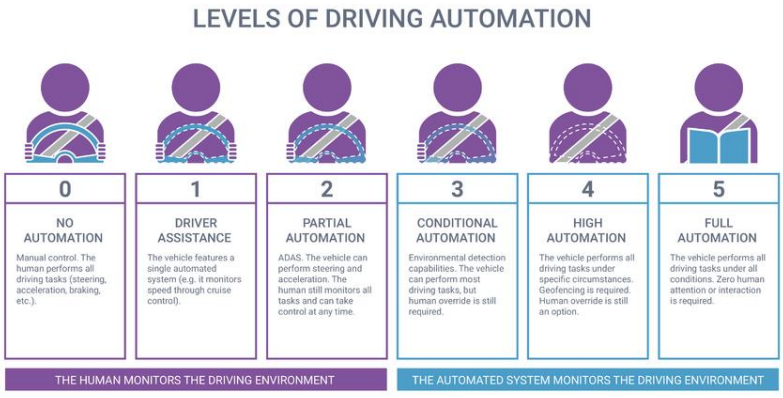

Figure 1.1 Scheme of the different levels of automation.

Driving automation is differentiated into 6 levels from 0 to 5. In level 0 there is absolutely no automation in the car, not even a cruise control system. As it escalates to 5, more and more features appear until the car is fully autonomous in any circumstance. By any circumstance, it is understood even driving in a trail in middle of nowhere. As it can be seen, the technology is nowadays far from reaching that level of automation although there are some systems that get very close. On the one hand, Tesla only reaches a level 2 autonomous driving as it only reaches the ability of steering, accelerating and braking by itself and, hence, an attentive driver is needed. On the other hand, Waymo, with is taxi service called “Waymo One”, has reached a level 4 automation service as there is no driver in their taxis. However, for legal reasons there is sometimes an employee of the company in the driver’s seat although it is not needed. And this is also one of the main curbs to the development of the technology, as legislation is several steps back in comparison to the technology and politicians are not leaving the technology and the market to develop freely. It is true, though, that there are many philosophical and ethical problems to be solved in the responsibility over the car acts and the consequences of this new breakthrough.

In any case, investigation is also being made in many universities and research centres all around the globe with promising results. In fact, the Universidad Carlos III de Madrid has developed and already deployed a level 5 autonomous driving minibus in the natural park of Timanfaya in the Canary Islands, Spain. In collaboration with the government of the region, this project has been able to occur in such a challenging environment as a volcanic island with a very complicated orography and a very delicate environment as it is a National Park. This is why, the autonomous driving technology is always linked with electric mobility, that is a huge benefit for the healthcare of the citizenship.

The scene is, in any case, very promising and it is a fact that the mobility market is very close to a breakthrough that will change completely how people travel and move around. With this purpose, this thesis works on proving a solution to GNSS systems that can help in many ways to overcome this fascinating challenge.

Figure 1.2. Level 5 autonomous minibus developed by Universidad Carlos III de Madrid.

1.2 The problematics

On the business side, the main problems that the technology suggest are the common ones that appear when a new market and technique is being developed. It is needed a maturity in the hardware and software development in order to reach scalation levels that make autonomous driving much cheaper because the work is still in progress.

On the legal side, as it is been said, politics and legislators are going the easy way banning autonomous cars instead of trying to understand and permit this technology and its market to develop. It is true though, that cities like Phoenix in the USA allow companies like Waymo to operate in their streets however it is not a general case. In fact, it is known that Tesla has had to make changes in their Autopilot system because of legislation .

On the technology side, the main problem that it is encountered is that the autonomous technologies that exist can only perform in closed areas. This is called “geofencing”, that is delimiting an area in which the autonomous car can drive because it is where it has been trained to do so. Hence, the adaptability of the technology is compromised in many ways. Tesla’s Autopilot is not geofenced, but it is also only applicable to motorways in certain standardised circumstances. This happens because the systems are built to just react to stimulus but do not provide a wider control of the situation itself. For this purpose, the GNSS technology can work very well as it complements the real-time short-range information with a navigation perspective that can help create this wider perspective of the different situations. However, this technology has some disadvantages that are not giving it a main role in the industry.

1.3 The GNSS problematic

Once given a wide perspective on the scene in which the thesis is developed, it is time to introduce the problems and opportunities that the GNSS technology provide to autonomous driving in vehicles.

As a brief introduction it is important to explain what a GNSS systems is and how it works. GNSS stands for Global Navigation Satellite System that are the ones that provide with geo-spatial positioning worldwide. The GPS (Global Positioning System) is only one type of GNSS system that is owned by the US government and operated by the space force of that country. Galileo is the European option that was launched in 2011 and that competes with the north-American one.

The technology basically works with a receptor that communicates with satellites orbiting the earth (a constellation of satellites) in order to determine the position of it. All of them are synchronised with atomic clocks that provide an accurate measurement by computing the time delay between the emission of the signal in the satellites and the reception of the signal in the receiver from at least 4 different satellites. Therefore, here is introduced one of the main constraints for GNSS navigation: when the receiver does not have enough satellites at sight to be able to have a sufficiently accurate measure that is valuable for an autonomous car. There are several driving environments in which these constraints are critical such as cities (with urban canyons), forests or other geographical features that hamper satellite visibility. This is something that can be proven in many day-to-day situations like driving with a car below a bridge in a motorway.

In this case, the GNSS receptor in the car loses visibility for one or two seconds and that results in many wrong measures in the localisation of the car. Furthermore, when driving in an urban environment such a city there are many streets in which the buildings are high enough to prevent the receiver to have visibility of enough satellites to have a reliable measure.

These situations are easily perceived in the study material in which the thesis stands. This work is developed based on the optimisation of a trajectory made by a car between Leganés, Spain and Madrid, where many urban features appear and cause errors. In the following figures it is shown the original trajectory as well as some singular situations that give a graphical definition of the errors that the receiver output. As it can be seen, the trajectory suddenly has points that stand outside the logical trajectory that it is described and makes the whole measure useless for an autonomous driving application as the accuracy is not reliable enough. The thesis will work over 6 singular points that are shown in the Figure 1.3. that identify 6 different critical situations in a GNSS system and that will be solved with the different models that will be developed through the different chapters of the thesis.

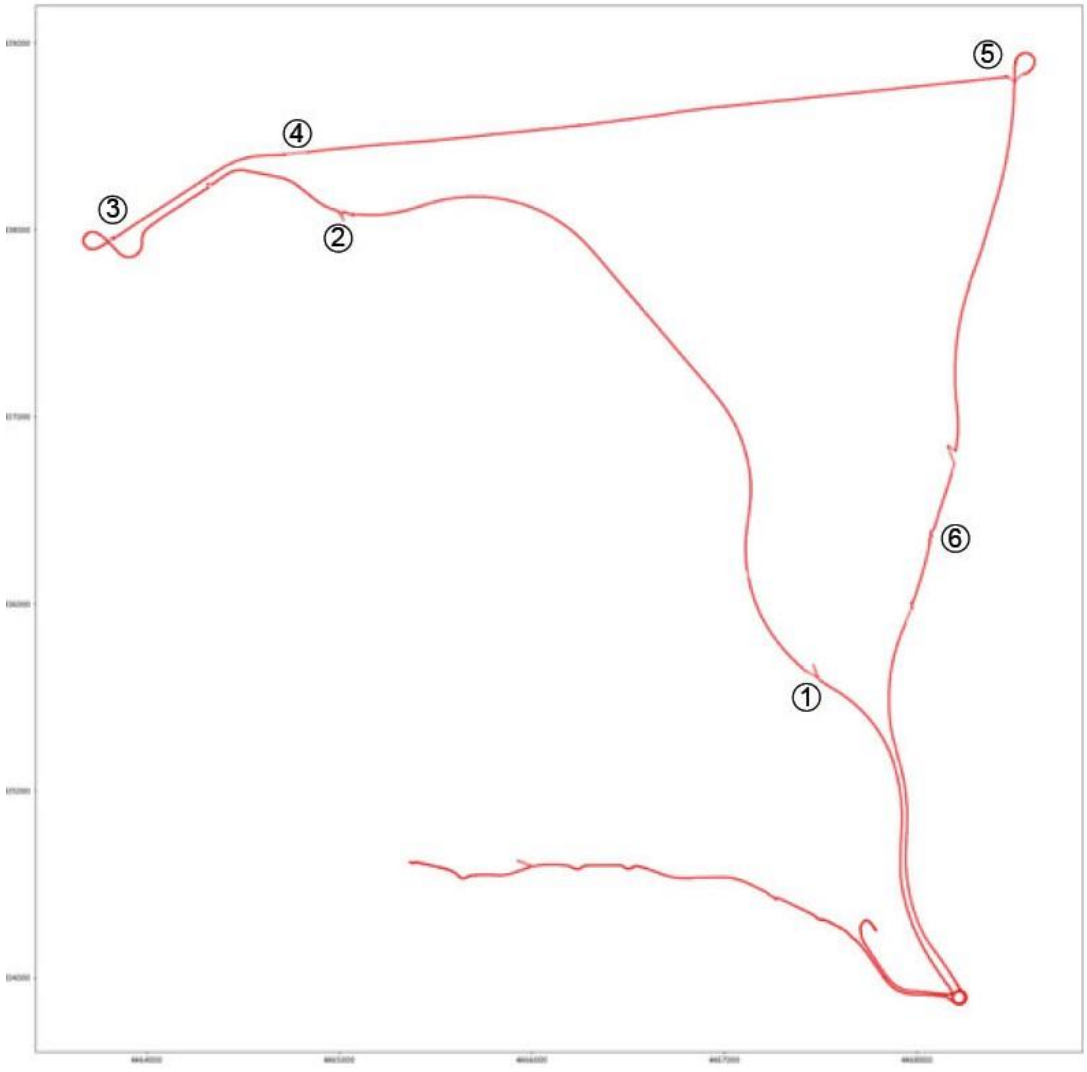

Figure 1.3. The full car trajectory and the 6 singularities to be studied.

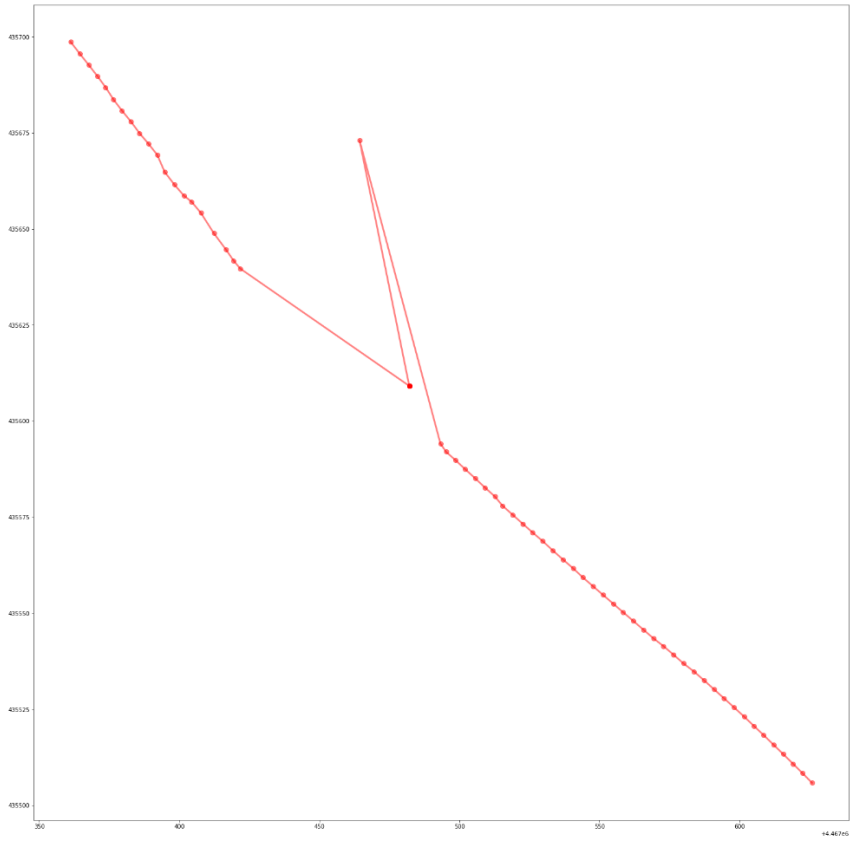

Figure 1.4. Singularity number 1.

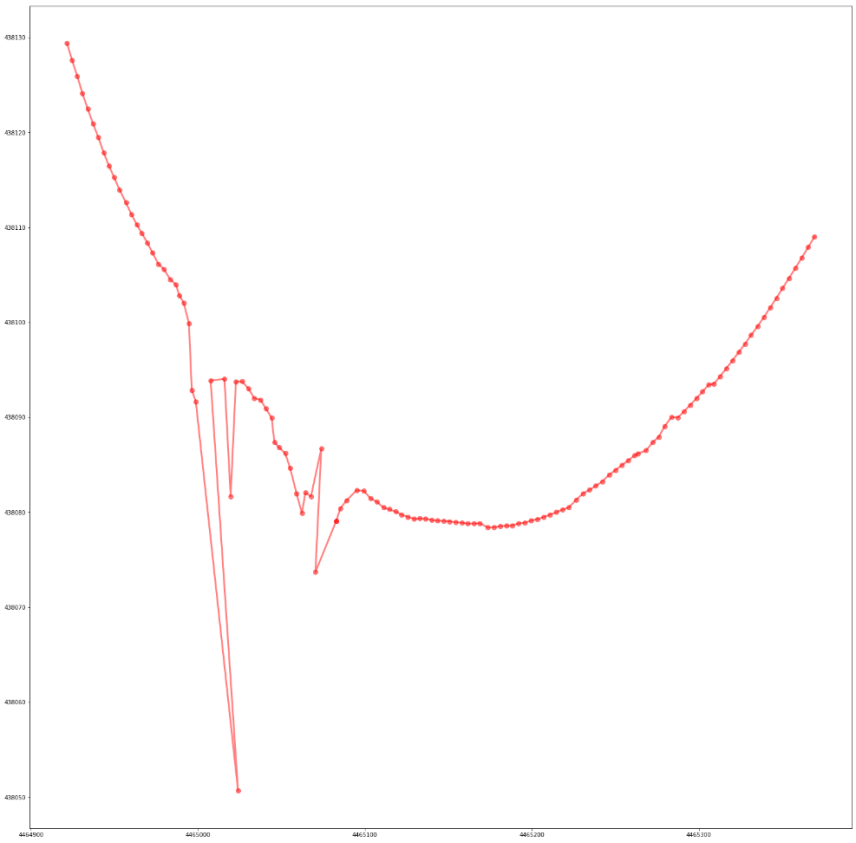

Figure 1.5. Singularity number 2.

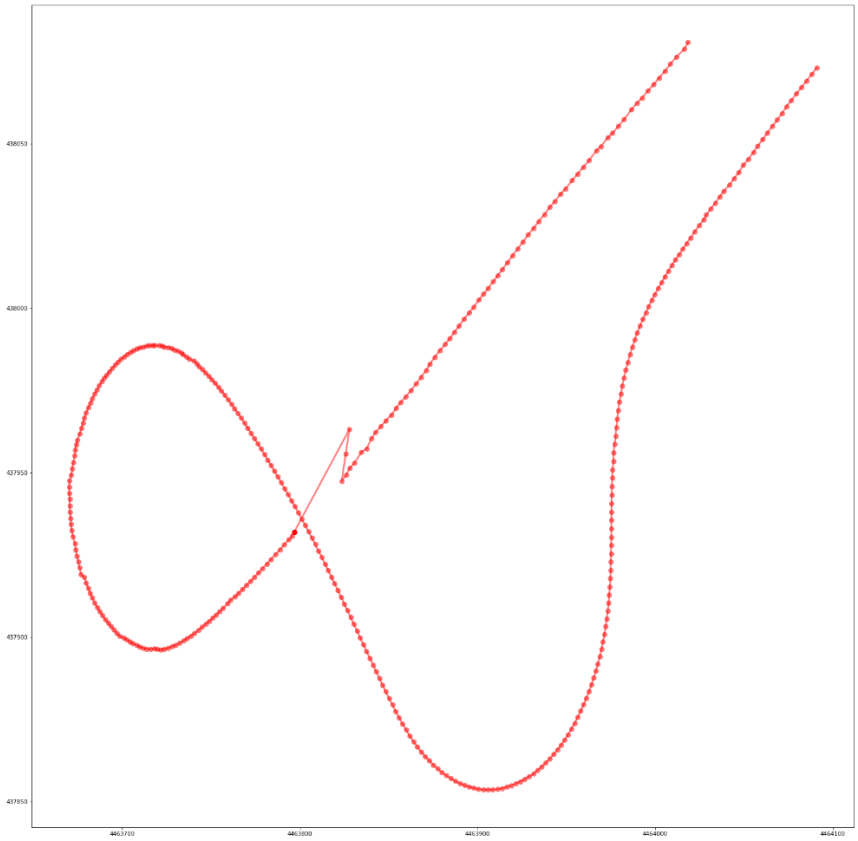

Figure 1.6. Singularity number 3.

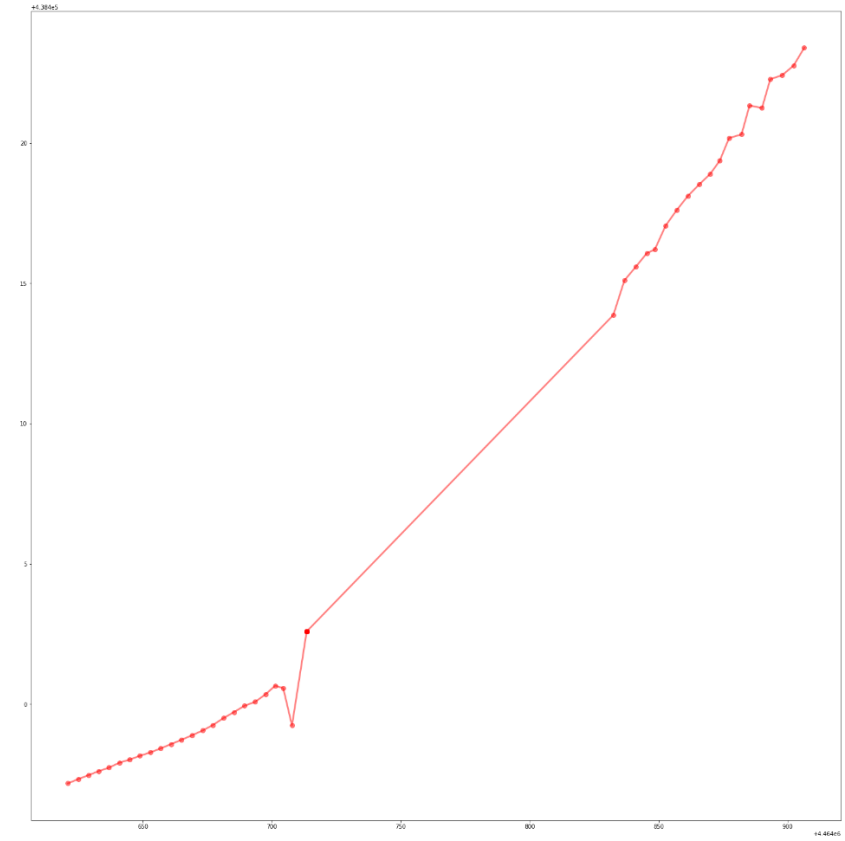

Figure 1.7. Singularity number 4.

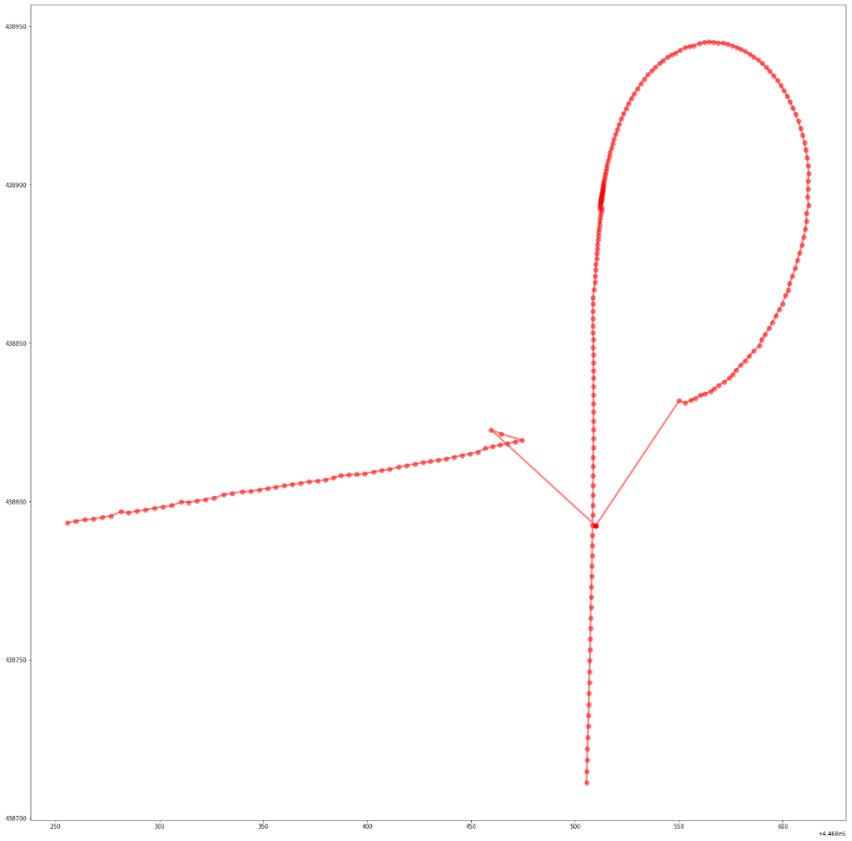

Figure 1.8. Singularity number 5.

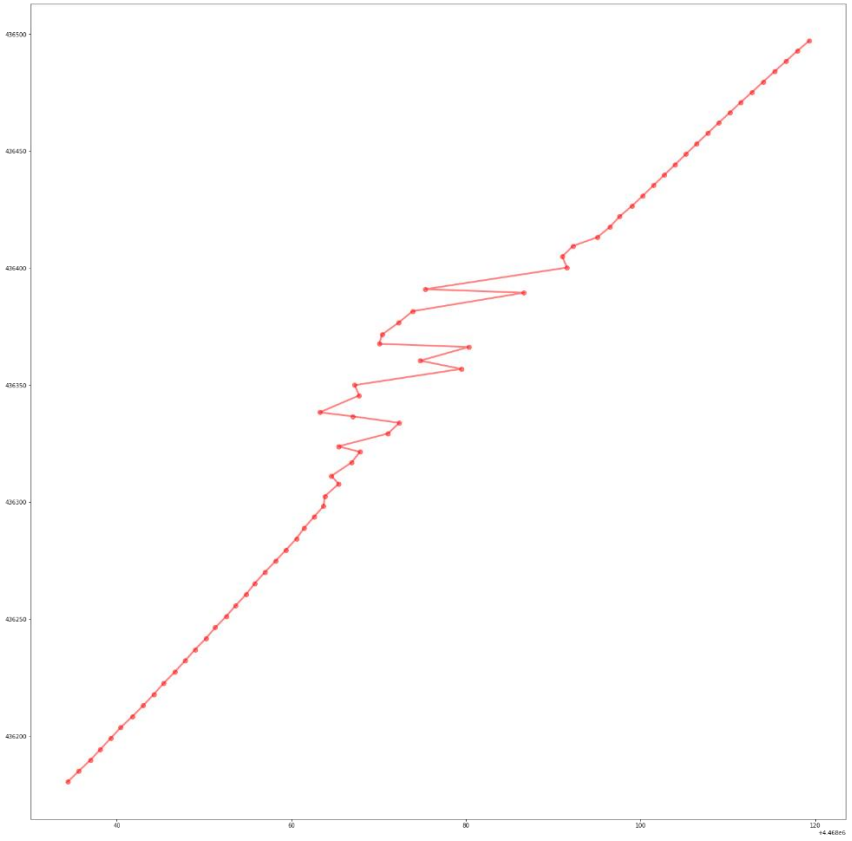

Figure 1.9. Singularity number 6.

2 STATE OF THE ART

In this chapter it is going to be presented several scientific papers that study the different perspectives, techniques and technologies that play a role in this thesis. All of them are recent works that will provide with a cutting edge vision of the environment of the thesis.

2.1 A fully automatic approach to register mobile mapping and airborne imagery to support the correction of platform trajectories in GNSS-denied urban areas

This paper is written by Phillip Jende, Francesco Nex, Markus Gerke and George Vosselman on a solution for trajectory correction in urban areas where GNSS is not an option as the several features make it impossible or too hard for this kind of systems to work. The present paper woks from a quite different perspective as it does not implement any optimisation of prediction algorithm like the Kalman Filter but uses imagery as the main source.

The proposed solution to correct the errors that appear in this type of situations is based on the correspondences between airborne nadir images that is specifically used for correcting the mobile mapping imaging data. In order to make this concept clear, mobile mapping is the process of collecting geospatial data from mobile vehicles that are equipped with several photographic, radar or laser sensors among many others. The solution can correct the orientation as aerial images do not output errors when in an urban area because the ground points and the several methods established to provide reliable data are consistent.

The main downsides of this technique are the corrections that must be made in scale, perspective or content that a priori very different and for this reason several traditional feature extraction and matching techniques fail. However, the implemented method, by focusing on common and clearly distinguishable features, it is capable of reach an accuracy close to 98%. Nonetheless, the paper does not cover the entire adjustment procedure and it is left for future works with the promise of achieving an accuracy of decimetres.

2.2 Quantitative analysis of GNSS performance under railway obstruction environment

This paper is written by Debiao Lu, Shuxian Jiang, Baigen Cai, Wei Shangguan, Xiankai Liu and Jin Luan and provides an evaluation of the GNSS performance in low visibility environments.

This paper provides a rich and updated view on how geographical and urban features affect the performance of GNSS systems. This technology is important in the railway sector because it is used for providing customer services as well as timing and safety applications that are critical. When entering a tunnel or an urban canyon, the GNSS signal is lost, and this could lead to critical situations if an emergency occurred. Hence,

it a critical situation in which a perfect performance of these systems is needed. Their study is based on a model of the Qinghai-Tibet line that recreates the scenario and the different constraints that appear in order to make a physical and mathematical analysis of the performance of the GNSS sensors and try to understand possible improvements that could be made in order to ensure a secure and stable connection during the trip.

2.3 ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras

This work is presented by Raúl Mur-Artal and Juan D. Tardós and presents a Simultaneous Localization And Mapping (SLAM) system for loop closing, map reuse and relocalisation capabilities. The purpose is to be used in stereo, monocular and RGB-D cameras.

The solution can be implemented in a wide variety of environments such as industrial robotics to commercial cars. It provides with a real time accurate trajectory estimation that is based on a bundle adjustment using monocular and stereo observations. As it a SLAM technology, there is a lightweight localization mode that makes it possible to track unmapped regions and find matches with already known map point allowing a localisation without any drifts. It is able to provide one of the most, if not the most accurate SLAM system that exists.

2.4 Scaling parameters selection principle for the scaled unscented Kalman filter

This paper is written by Nie Yonglang and Zhang Tao and provides a solution for resolving the scaling factors that affect the Unscented Kalman Filter.

There is nowadays a controversy over the most adequate adjustment of the scaling parameters when constructing a model based on an Unscented Kalman Filter. Different sources do not agree on a law or procedure to determine these values and this is usually done by manual adjustment using trial and error. Their selection principle stands in seven steps:

1. Initialise mean and covariance.
2. Estimate a set of scaling parameters $\alpha = 1$, $\beta = \beta_c$ and $\kappa = \kappa_c$.
3. Calculate the sigma points.
4. Calculate the updated state mean and covariance.
5. Compute α_k.
6. Let $k = k + 1$ and repeat steps 3 to 6.
7. Compare the covariance that is estimated with the UKF default one, select the smaller covariance and use the associated mean as the valid result.

The mathematical explanation is not put in this summary as it is very long. However, it is highly recommendable to read the explanation in order to fully understand the technique that it is used.

Finally, it is important to note that there is no performance improvement intended but only establishing a procedure for estimating the scaling parameters of the unscented transform for the construction of an Unscented Kalman Filter.

2.5 Empirical evaluation of vehicular models for ego motion estimation

This is a work made by Robin Schubert, Christian Adam, Marcus Obst, Norman Mattern, Veit Leonhardt and Gerd Wanielik.

Their study has the objective for solve one of the main problematics with motion estimation in intelligent vehicles. For this purpose, they review the most common vehicles mathematical and physical models and face them to different scenarios. Then , they study the suitability of different models in order to obtain a result that is the most optimal option. The main purpose is to provide some guideline son how to proceed when choosing the best model for motion studies of a vehicle.

On the behalf of the thesis, the most important output of the paper is that noise that the covariance presents, can be done by a matricial model that considers that it has a constant value of 8.8 m/s^2.

3 DATASET

Before diving into the technological solution to the errors that occur in the GNSS system, it is important to understand the data with which the models work.

The data is provided in a .txt file in which appears a velocity value in the #BESTVELA log and a position value in UTM in the #BESTUTM log for each point of time. Both of them output many data that is complementary to the mere velocity and position in UTM that also explains the configuration of the receiver. Some of these configuration and state fields that are important to comment are:

- **FINESTEERING** that denotes that the time is fine set and is being steered.
- **SOL_COMPUTED** stands for the solution computed status of the solution. During the sample will change where inconsistencies appear. The main different values that will appear are: **INSUFFICIENT_OBS** that means the observations are insufficient and, hence, there is a signal loss and **COV_TRACE** that appears when the covariance trace exceeds the maximum of 1000m. This last status will appear when the signal is very poor but it is not lost.
- **PSRDIFF** that denotes the type of position or velocity data being recorded being in this case a solution computed using pseudorange differential corrections. This field will change eventually, to **SINGLE** when the solution is calculated using only data given by the GNSS satellites and to **DOPPLER_VELOCITY** only for the velocity being calculated using instantaneous Doppler.

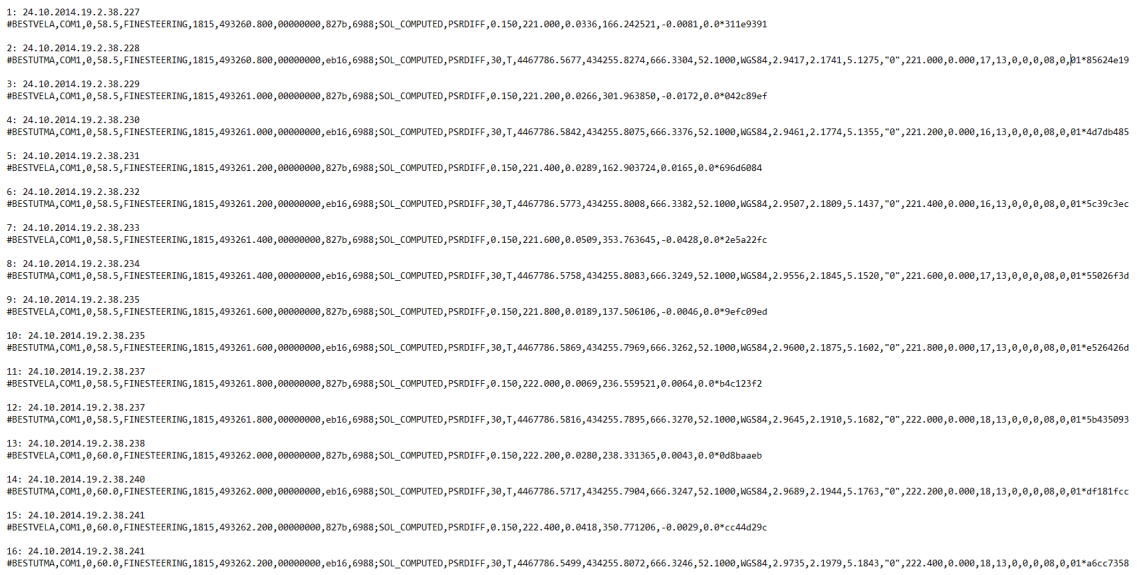

Figure 3.1. Extract of the dataset showing 8 points of time.

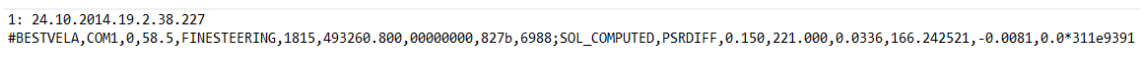

Figure 3.2. Zoom on a #BESTVELA measurement

It is also important to explain what UTM (Universal Transverse Mercator) coordinates are and how they are related to the normal scale used for geo-positioning in degrees, minutes and seconds. The main idea is that the earth is divided into zones that are not necessarily symmetrical and are a total of 60 parallel to the imaginary longitude lines. Then, it is made another division in another 26 zones each one corresponding to a letter of the alphabet from A to Z. With all this, the result is a grid in which a number followed by a letter designates a specific zone. Moreover, the numerical values for the coordinates need to be specified if it is for the north or south hemisphere (in this thesis it is for the north hemisphere) and, hence are referenced to the equator that is the 0 value for the y axis and the x value varies within the delimited zone.

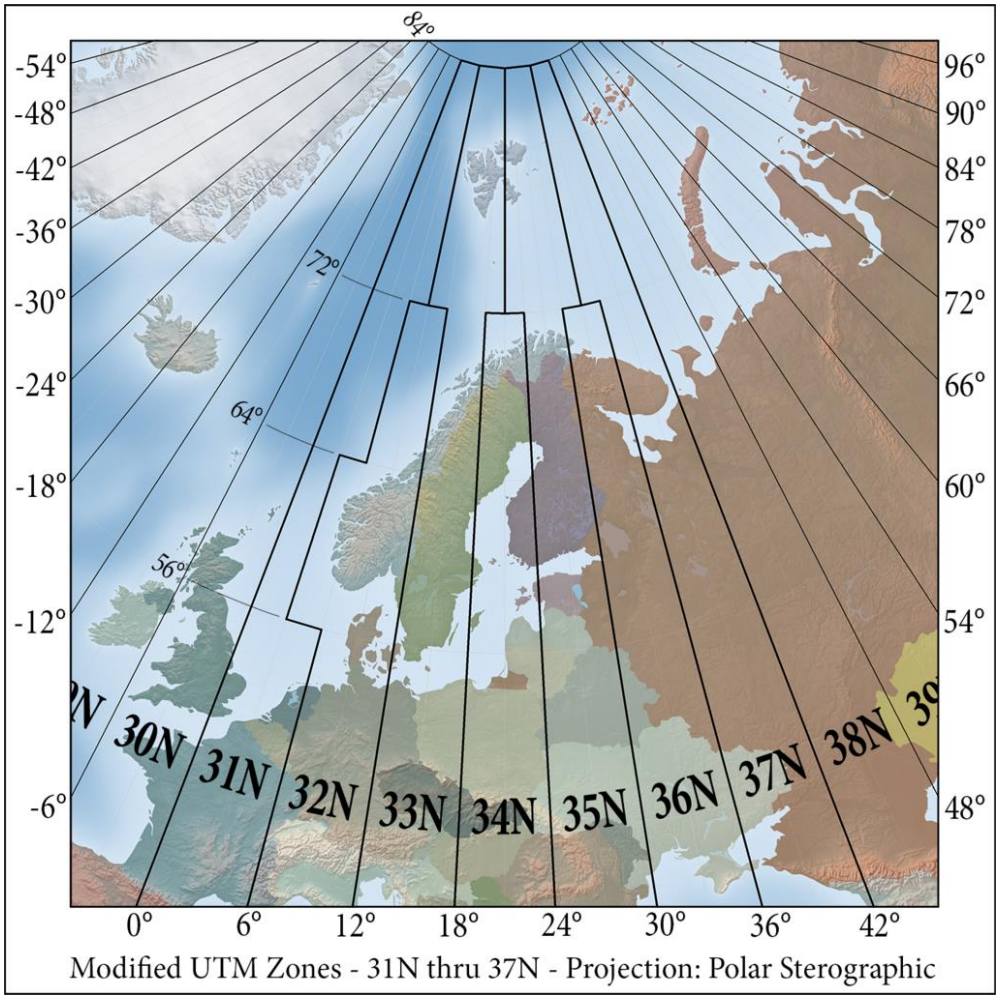

Figure 3.3. Representation of how UTM zones are extracted from the latitude-longitude map.

All the data given by the GNSS receptor, is treated using Python programming. In fact, everything in this thesis except for the MATLAB toolbox used in the Unscented Kalman Filter Model, will be made with Python. From reading the .txt file, extracting the desired data for the model, the model construction as well as all the mathematical and algebraical computations and the results visualization.

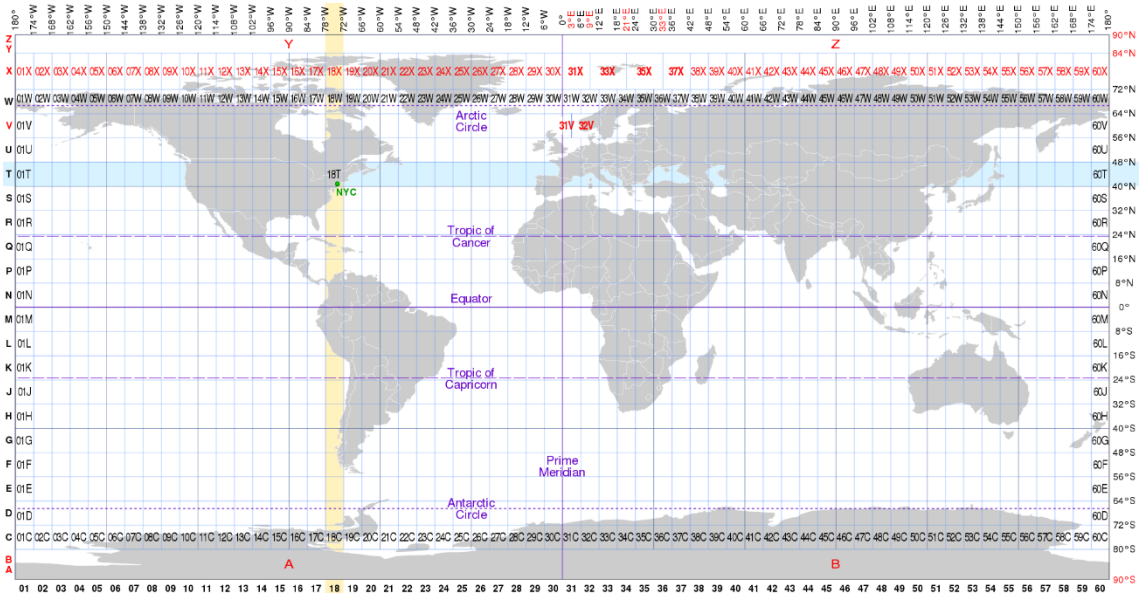

Figure 3.4. UTM grid for the world map.

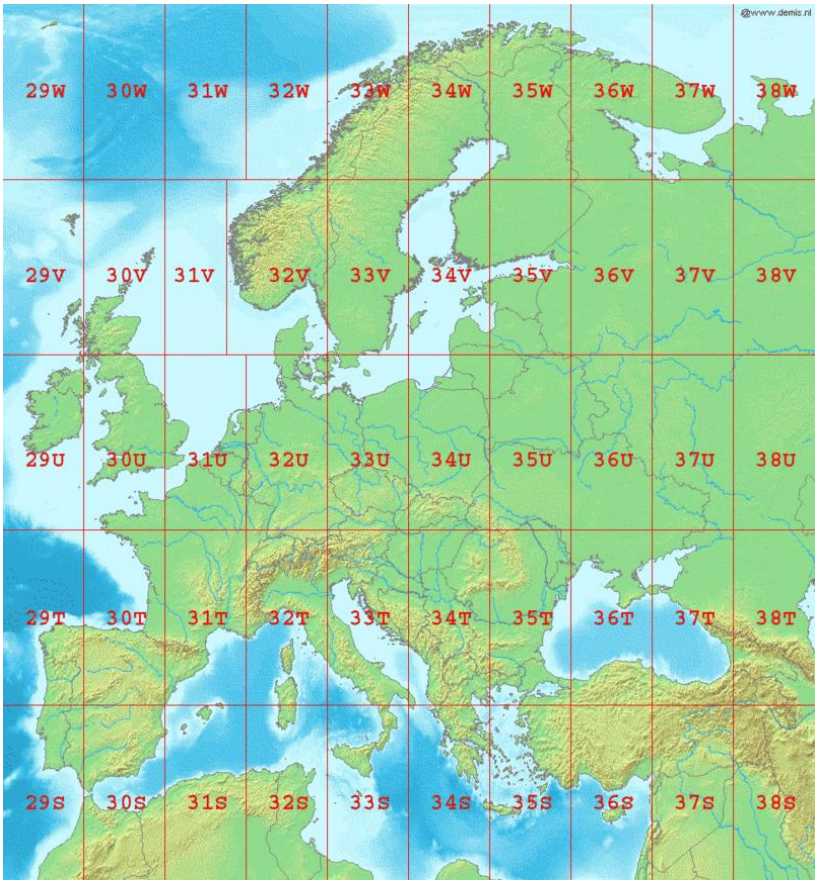

Figure 3.5. UTM grid for Europe's map.

4 KALMAN FILTERING

Now that it is been provided a sufficient wide perspective of the problematic that the thesis works on, it is time to dive into the solution proposed and understand how Kalman Filters work and its application to the study case.

4.1 The solution

As it is been said before, the main purpose of this thesis is to find a method to solve the inconsistencies of GNSS live location when driving a vehicle. Geographical and, especially, urban features are the main cause of the errors that a GNSS-based location system presents. The solution of this, will give a better navigation experience that can permit the evolution of both autonomous vehicles but also other services and technologies such as navigation. It is not rare that, while driving, a normal person has problems with its navigation system because of line misunderstanding or even location confusions between two roads that are very close one to another (something that happens more commonly in big cities where many different roads converge and diverge).

For this reason, it was needed a technique that unified performance, reliability and simplicity and the result was selecting Kalman Filtering. The system load (the amount of computational work that it is needed) is not very high in the case of this kind of algorithm as it is an iterative solution that in based on the dynamic behaviour of the system. The performance, as it is going to be demonstrated in the following pages, is very precise and is able to solve critical situations when the satellite signal is interrupted or drastically reduced. Finally, the simplicity of the solution makes it easier to be developed, adapted and deployed in the GNSS system and also reduces de computational load.

4.2 What is a Kalman Filter?

A Kalman Filter is an iterative algorithm that makes a prediction of the future state of a system. This prediction is made through some mathematical expressions that represent the behaviour of the system and are specific of each case study. Once made a prediction, the result is updated and adjusted with a reliable reference that comes from a sensor and provides some information of the evolution of the system. Another important characteristic is the quickness of Kalman Filtering. This is due to its low level of complexity makes it computationally light and provides results in a very fast way.

The idiosyncrasy of the method can be seen in Figure 3.1. in which an example based on the case study is established. In this example there is a car in the present time designated with time t. The Kalman Filter, based on a dynamic model of the car and the velocity sensor reference, will make a prediction on where the car will be in the next “moment” designated with t+1. It is also important to note that, as it is inferred in the picture, that Kalman Filters work on discrete time and not in continuous time. Predictions and updates work on the current and in the immediately next “point of time”. Once finished the current prediction, the filter moves forward one “point of time” and performs the same computations. This is repeated for the whole sample.

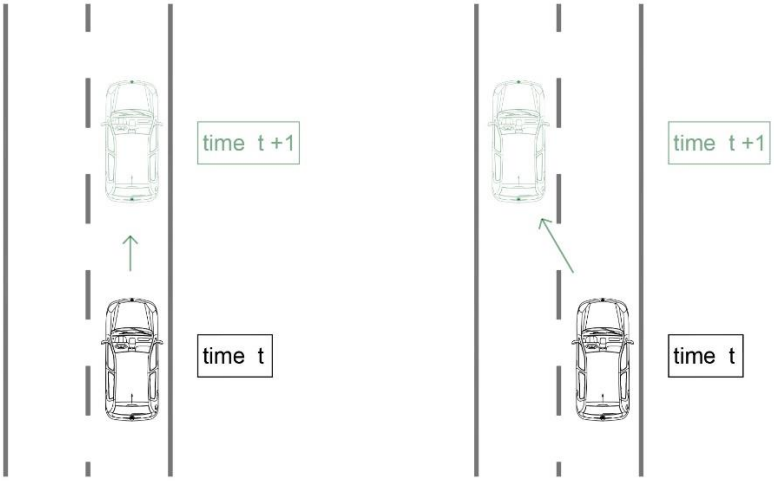

Figure 4.1. In time t it is made a prediction on where the car will be in t+1 based on the dynamics of the car.

4.2.1 Core theoretical concepts

Firstly, there are three main theoretical concepts that must be understood before going with the mathematical explanation and that are common to all Kalman Filters (or mostly as there could appear slight differences in some). These are:

1. Understanding what a normal or gaussian distribution is and its application to the "points of time".
2. Understanding how prediction and iteration operate and the workflow of the algorithm.
3. Understanding what the Kalman Gain is.

So, in first place it is going to be discussed the concept of normal or gaussian distribution. In the field of probability theory, this kind of distribution is a continuous probability distribution, what means that it is cumulative and absolutely continuous. It is applied for real-valued random variables that is the exact behaviour of the "points of time" that are used in the Kalman Filter model. The probability density function is:

$$f(x) = \frac{1}{\sigma \cdot \sqrt{2\pi}} \cdot e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \tag{4.1}$$

μ: mean of the distribution (point of time in Kalman Filtering).

σ: standard deviation.

The equation above describes the gaussian distribution in one dimension. However, in the case to be studied in this thesis both "x" and "y" coordinates present normal

distributions for their values in each point of time. This is important while understanding the next plot that describes the distribution form graphically:

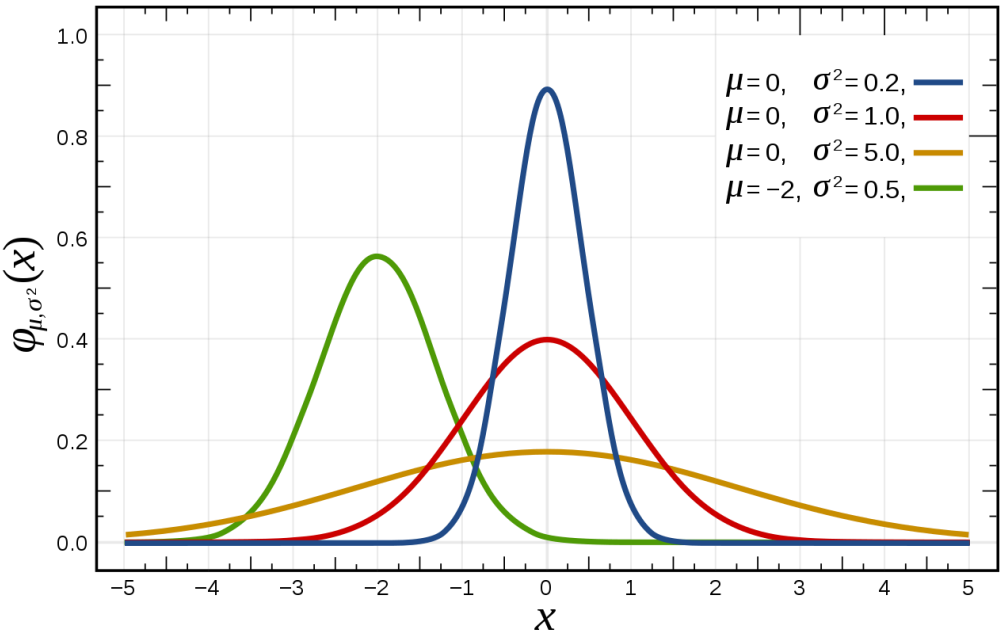

Figure 4.2. Plot of different Gaussian distributions showing the influence of μ and σ².

Once it is understood what a Gaussian distribution is, it is time to explain in depth how the two main steps of the filter work. This is the core concept of the algorithm and it is very important to understand it properly. The Kalman filter follows an iterative structure that consists of three steps:

1. Current State: It is the state in which the algorithm is at the beginning of the iteration before any changes have been made.
2. Predicted Step: It is the state that results from the projection of the current state through the mathematical equations that explain the behaviour of the system.
3. Updated Step: It is the final step in which the prediction is adjusted with the Kalman Gain and the sensor measurement. After this step, the iteration is finished.

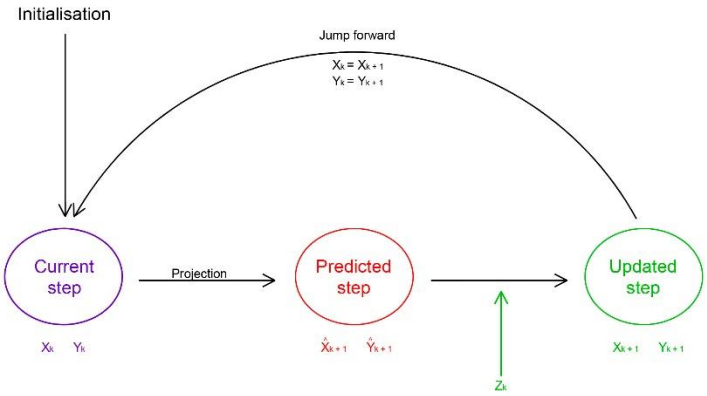

Figure 4.3. Step Diagram of the Kalman Filter Workflow.

After these three steps are finished, the computed state becomes the current state and the filter moves forward with another iteration in the next point of time. It is important to know that, when the filter starts (the iteration number 1), it is needed an initialisation with some given values for the different variables that actuate. This workflow is shared among the different variations of Kalman Filtering such as Extended Kalman Filter (EKF) or Unscented Kalman Filter (UKF).

Finally, it is fundamental to understand what the Kalman Gain (K) is. The Kalman Gain is the parameter that allows to adjust the model in the update step. Without this parameter, the predicted model would not have any form of proving if the mathematical projection is absolutely correct or not, considered that the points have errors (following a gaussian distribution as explained before).

4.2.2 Mathematical explanation of the Kalman Filter

Now that the main theoretical concepts are clear, it is time to continue with a mainly mathematical but also conceptual explanation of the Kalman Filter. For this purpose, the schema of an iteration will be followed explaining all the mathematics that rely behind it.

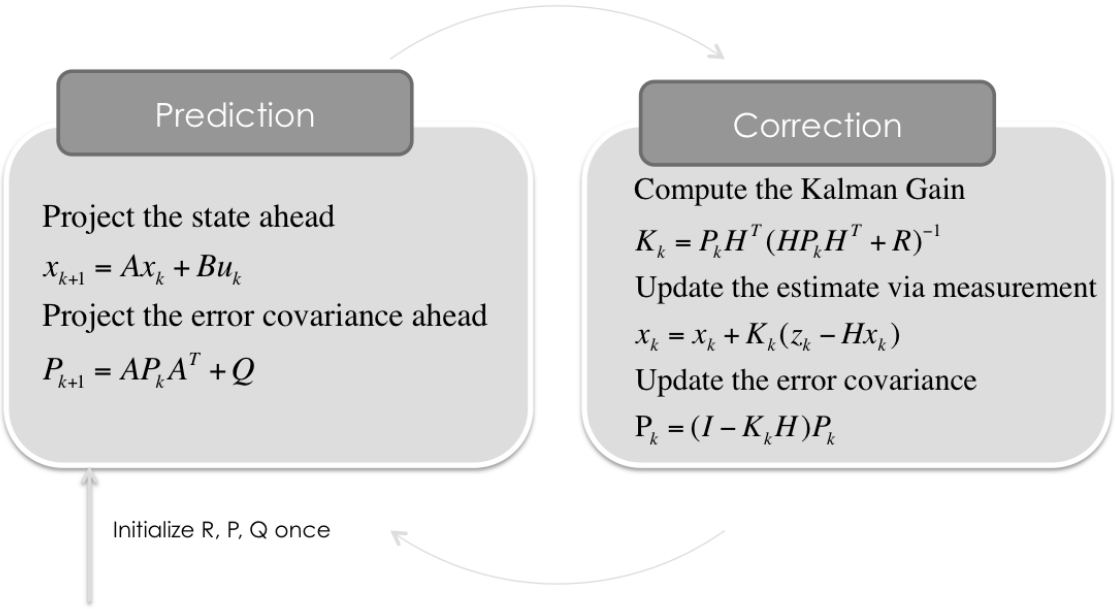

Figure 4.4. General schema of the equations that describe the Kalman Filter.

The Figure 4.3. shows the different equations and variables that take part both in the prediction and the correction step. The variables that appear in the prediction step are:

- $\mathbf{x_k}$ is the state matrix that records the information to be tracked from an object. It has a *(n x 1)* shape.

- **A** is the transition matrix of the dynamic model. It is the "matricial translation" of the function that projects the state to the next one based on the mathematical model that explains the behaviour of the system. It has a (*n x n)* shape.
- **B·u_k** is a bias parameter that introduces noise to the projection in order to obtain a better result. Is one of the parameters to be adjusted manually. B has a (*n x n)* shape and u_k a (*n x 1)* shape.
- **P** is the covariance matrix. It is the one that has the information of the error of the estimations made with the filter. It has a (*n x n) shape.*
- **Q** is the process noise covariance matrix. It is another matrix that must be adjusted manually to the model to obtain the maximum performance. It also prevents the covariance to be equal to 0. It has a (*n x n) shape.*

Once explained the variables of the first part of the filter, now it is going to be discussed the new variables that appear in the update step of the algorithm:

- **K** stands for the Kalman Gain. It is a crucial concept that has already been explained in the previous section. It is the parameter that adjusts the prediction in the update step. Its shape depends on the matrix H that will be explained hereunder.
- **H** is the measurement observation matrix. It relates the state matrix with the measurement matrix. For this to be understood, imagine a situation in which the measurement of the sensor does not provide straightforward the information that is in the state matrix. This could be the case of a velocity (v) sensor that outputs 10·v due to the configuration of it. In that case, the measurement matrix is in charge of converting the measure to one that is useful for the model by modulating its values. Its shape depends on the measures extracted from the sensor but will always have *n* columns (same columns as variables or rows has the state matrix **x**).
- **R** is the noise measurement matrix. It is another variable that should be adjusted manually and can be based on data given by the manufacturer.
- Finally, **z**, is the measurement matrix that provides the reference data from the sensor to perform the update. It has as many rows as variables measured with the sensor and one column.

It is important to note that the manual adjustment of the different matrices in the algorithm must be done before the filter starts and remain unchanged during the iterations.

4.2.3 The Unscented Kalman Filter (UKF)

UKF is an "evolution" of the Kalman Filter to improve its performance in real life applications. The main problem that vanilla Kalman Filter presents is that they rely on the assumption that the data is absolutely gaussian and that the equations that describe the systems and its behaviour are linear. However, this does not happen in most cases

and real life applications. In order to overcome this huge limitation, two “evolutions” appeared: the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF).

This thesis focuses on the UKF because its applicability to the case study is much better because of the technique used for managing the non-linearity. Whereas the EKF linearises the equations obtaining an approximation based on one point (the reference point for the linearization), the UKF computes several points called “sigma points” and based on that constructs the projection of the gaussian when going through the non-linear function.

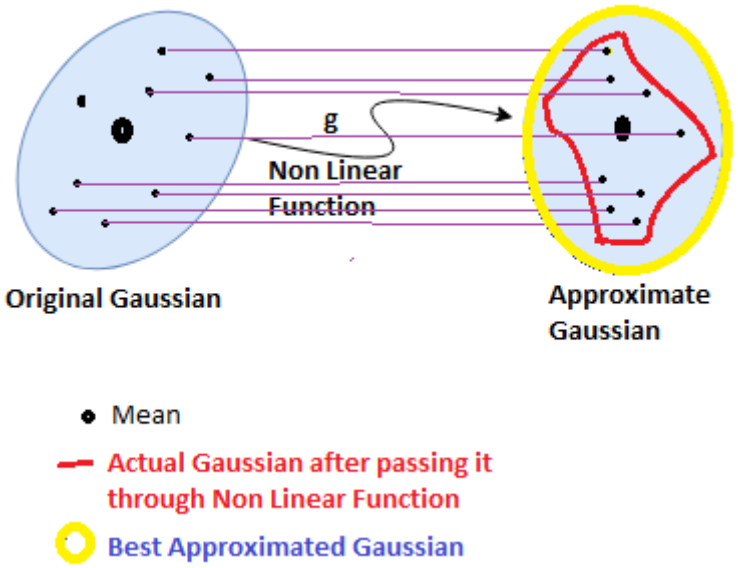

Figure 4.5. Graphical explanation of the sigma points usage in gaussian approximation after projection.

As it is inferred by the previous paragraph, EKF is less precise than the UKF. Nonetheless, this lack precision results in a constraint that directly discards the EKF for the case study. This issue is that error propagation becomes very high and for long samples, as the one used in this thesis that has over 7000 points, produces errors eventually that are unacceptable. Therefore, although UKF might require more computational power, it is capable of overcoming error propagation through the sample.

Therefore, the is the main theoretical concept that is added by the UKF and should be properly understood before going with the mathematics is the Unscented Transform. It is also important to make it clear that prediction, update and iterations idiosyncrasy remains the same as in Kalman Filter and so does the Kalman Gain. The Unscented Transform is constructed through 3 steps:

1. Compute Set of Sigma Points.
2. Compute Wights of Sigma Points.
3. Compute Mean and Covariance of the approximate Gaussian.

Therefore, in first place, the sigma points (which are the small black points shown in Figure 3.4.) are computed as follows considering that there will be 2n + 1 points (n denotes the dimensions of the system):

$$\chi^{[0]} = x \tag{4.2}$$

$$\chi^{[i]} = x + \left(\sqrt{(n+\lambda)\mathrm{P}}\right)_i \qquad \text{for i = 1,…,n} \tag{4.3}$$

$$\chi^{[i]} = x - \left(\sqrt{(n+\lambda)\mathrm{P}}\right)_{i-n} \qquad \text{for i = n+1,…,2n} \tag{4.4}$$

Where:

- **X** denotes the Sigma Points Matrix. This matrix has a (n x 2n+1) shape where each Sigma Point corresponds to a column of the matrix. Therefore, $X^{[0]}$ will be the first column of the matrix, equation 3.3 will compute the second column until the value “n” and then, from the column “n+1” until “2n”, is computed with equation 3.4.
- **x** denotes the mean of the Gaussian. This “x” is exactly the same as the one that appeared in the Kalman Filter and is also the state matrix of the filer. Its shape remains the same as it is a (n x 1) matrix.
- **n** denotes the dimensionality of the system and it is a number.
- **λ** is the scaling factor. It is computed as it is shown in the equation 3.5. However, there is controversy over how the parameters α and κ should be adjusted in order to perform correctly the unscented transform in the different scenarios. Some studies suggest that the scaling factor can be adjusted directly as it shown in equation 3.6, neglecting both α and κ. Moreover, there is another parameter β that appears in the second step while computing the weights of the Sigma Points and has the same controversy over how it should be adjusted. This thesis is going to solve this problem based on the paper written by Nie Yongfang and Zhang Tao called “Scaling parameters selection principle for the scaled Unscented Kalman Filter”.

$$\lambda = \alpha^2(n+\kappa) - n \tag{4.5}$$

$$\lambda = 3 - n \tag{4.6}$$

- **P** is the Covariance Matrix that remains the same as in the previously explained Kalman Filter. Its shape is (n x n).

After computing the Sigma Points matrix, the next step is calculating the weights associated to each Sigma Point as:

$$w^{[0]} = \frac{\lambda}{n+\lambda} \tag{4.7}$$

$$w^{[i]} = \frac{1}{2\cdot(n+\lambda)} \qquad \text{for i = 1,…,2n} \qquad (4.8)$$

It is very important to note that the sum of all the weights associated to the different Sigma Points must be equal to 1.

In fact, the Unscented Transform is the tool that creates the state projection in the UKF. After performing the two first steps in which the Sigma Points and their weights are computed, the last phase consists on making the corrected projection itself computing the mean and covariance of the approximate Gaussian. In this step it is made at the same time the projection and the correction based on weights and Sigma Points as:

$$x' = \sum_{i=0}^{2n} w^{[i]} \cdot g(\chi^{[i]}) \qquad (4.9)$$

$$P' = \sum_{i=0}^{2n} w^{[i]} \cdot \left(g\left(\chi^{[i]}\right) - x'\right) \cdot \left(g\left(\chi^{[i]}\right) - x'\right)^T \qquad (4.10)$$

Where:

- **x'** is the predicted mean. Its shape stays unaltered being (n x 1).
- **P'** denotes the predicted covariance matrix. Its shape remains also unaltered being (n x n).
- **w** corresponds to the weights of the Sigma Points.
- **g(x)** is the non-linear function that describes the behaviour of the system.
- **X** is the Sigma Points Matrix.
- **n** is the dimensionality of the system.

Following both equations, the result of the state matrix "x" and the covariance matrix "P" comes from summing the 2n+1 Sigma Points after having gone through the non-linear function that describes the evolution of the system. With all this, the prediction phase is finished.

After the prediction step, the update phase takes place. As it is been repeated several times during the thesis, these two steps are the same for every Kalman Filter and the Unscented Kalman Filter is not an exception. Therefore, the correction or update step starts by taking one decision: computing again the Sigma Points based on the predicted version of them or keeping the Sigma Points that have been already computed. In this case, it is going to be taken into consideration that they are kept the same for the following calculations. The update step could also be divided in three different phases:

1. Compute the measurement space (based on sensor data).
2. Compute the Kalman Gain.
3. Update the predicted mean and covariance matrices.

Firstly, the measurement matrix is calculated based on the Sigma Points and their weights as:

$$Z = h(\chi) \tag{4.11}$$

$$\hat{z} = \sum_{i=0}^{2n} w^{[i]} Z^{[i]} \tag{4.12}$$

Where:

- **Z** is the result of the transformation of the Sigma Points into de measurement space. Its shape depends on the dimensionality of the data given by the sensor.
- **X** denotes the Sigma Points Matrix.
- $\hat{\mathbf{z}}$ is the measurement matrix and denotes the mean of weighted Z. After bringing the Sigma Points to the measurement state, it is computed the mean of them to output a one-column matrix that can be used in the following equations as the "adjusted" measurement matrix based on the Sigma Points calculation. Its shape depends on the measurements of the sensor.
- **h** is the function that is equivalent to the matrix H in the Kalman Filter. It transforms the computed state space to the measurement space from which the sensor data is received. Its shape is also dependant on the sensor data. However, it is important to note that the number of rows will be equal in h, Z and $\hat{\mathbf{z}}$.

The next step is computing the Kalman Gain. This is another of the core concepts around which the different Kalman Filters are constructed. Perhaps, this is the most important of them as is the parameter that ensures that the state projection is done correctly by adjusting the first prediction in the update stage. The computation of the Kalman Gain is done as follows:

$$S = \left[\sum_{i=0}^{2n} w^{[i]} \cdot \left(Z^{[i]} - \hat{z}\right) \cdot \left(Z^{[i]} - \hat{z}\right)^T\right] + Q \tag{4.13}$$

$$T = \sum_{i=0}^{2n} w^{[i]} \cdot \left(\chi^{[i]} - x'\right) \cdot \left(Z^{[i]} - \hat{z}\right)^T \tag{4.14}$$

$$K = T \cdot S^{-1} \tag{4.15}$$

Where:

- **S** denotes the Covariance in Measurement Space Matrix. It is a squared matrix whose dimensions are delimited by the dimensionality of the measurement.
- **Q** is the Measurement Noise Matrix. It has to have the same shape as the matrix S and its values must be manually adjusted.
- **T** is the Cross Co-relation Matrix between the measurement space and the predicted space. It is a matrix with *n* rows and the same number of columns as the dimensionality of the measurement.
- **K** denotes the Kalman Gain. It is a matrix with *n* rows and the same number of columns as the dimensionality of the measurement.

With the computation of the Kalman Gain, the second stage of the update step is finished. To make the final update of the iteration, the equations to be followed are:

$$x = x' + K(z - \hat{z}) \tag{4.16}$$

$$P = (I - KT)\, P' \tag{4.17}$$

Where:

- **x** is the mean or State Matrix.
- **P** is the Covariance Matrix.
- **x'** is the predicted mean.
- **P'** denotes the predicted Covariance Matrix.
- **K** is the Kalman Gain Matrix.
- **z** denotes the actual measurement coming from the sensor.
- $\hat{\mathbf{z}}$ denotes the mean in the measurement space. It is important not to confuse it with z. Conceptually $\hat{\mathbf{z}}$ is an expression of the measurement in a space constructed with the Sigma Points and weights computed for the iteration. On the other hand, **z** is the raw measurement that comes from the sensor.
- **T** is the Cross-Correlation Matrix.

After equations 3.16 and 3.17, the iteration is finished and the filter continues with the next point of time.

4.3 The Kalman Filter applied to the case study

In this thesis, there are three Kalman Filter models from which the two first are constructed with a standard Kalman Filter and the last one is an application of the Unscented Kalman Filter. All of them study the case of a car that drives in different environments in between the Spanish cities of Leganés and Madrid in one big sample. The data of was recorded by a FlexPak-G2 NOVATEL GNSS receptor that provides each 0.15 seconds a datum of velocity and UTM position along with many other parameters that are not object of study in this thesis although some of them are adjustments of the sensor that generate the desired data output. It is important to remember that UTM coordinates result from a projection in the plane of the usual coordinates that are the standard for geo-localisation.

Therefore, the State Matrix is defined for all three models as:

$$\bar{x} = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \end{bmatrix} \tag{4.20}$$

So that the dimensionality of the model is n=6 because the state matrix will keep track of position in both x and y axis and the velocity and acceleration in them.

The objective of this thesis is to correct the errors in the x-y positioning of the GPS as the output coming from the GNSS receptor is very inaccurate, especially in environments such as urban or while driving down a bridge over other situations. For this purpose, the measurements from the sensor that are reliable are the velocity in x and the velocity in y. Therefore, the Measurement Matrix is constructed based on these data such as:

$$z = \begin{bmatrix} \dot{x}_{sensor} \\ \dot{y}_{sensor} \end{bmatrix} \tag{4.21}$$

And will have a *2x1* shape, therefore determining the shape of dependant matrices such as H.

On the other hand, the Transition Matrix will depend on the model and it is going to be explained in depth in the following chapter. However, it is important to understand that in this matrix (in the UKF is the function **g**) is constructed based on a dynamical analysis of the car motion with the equations Uniform Rectilinear Motion (URM) and the Uniformly Accelerated Rectilinear Motion (UARM) for each of the two first cases. The assumption of an URM is applied in the first Kalman Model and the assumption of an UARM is applied in the second Kalman Filter and in the Unscented Kalman Filter application. It is interesting to remember the equations for both motion models in order to be able to make a deep analysis in the next chapter, when the three models will be explained in depth. For the URM the equations are:

$$x = x_0 + v \cdot t \tag{4.22}$$

$$v = const \tag{4.23}$$

Whereas the URMA is mathematically described as:

$$x = x_0 + v_0 t + \frac{1}{2} a t^2 \tag{4.24}$$

$$v = v_0 + at \tag{4.25}$$

With these simple, clear and straightforward equations, the performance of the Kalman Filter is raised because in this algorithm it is very important the balance between reliability and being computationally light and simple in order to achieve good results that can be applied to an application that corrects positioning error while driving, in live.

The rest of the Kalman parameters will be explained in the next chapter as they are highly dependent on the proposed model and change with it. However, it is important to have a clear perspective of the application of the Kalman Filter algorithm in the study case. The technology developed in the thesis is based on two inputs: UTM position and

velocity. The first throws incorrect points because of urban and motorway features and the second gives a reliable output. Hence, it is the position in x and y what is purposed to predict and to be corrected. The velocity data enters in the algorithm in the update phase in order to re-adjust the predictions made on the future position of the car. With main concept clear, it is time to advance to the explanation of the three Kalman Models proposed.

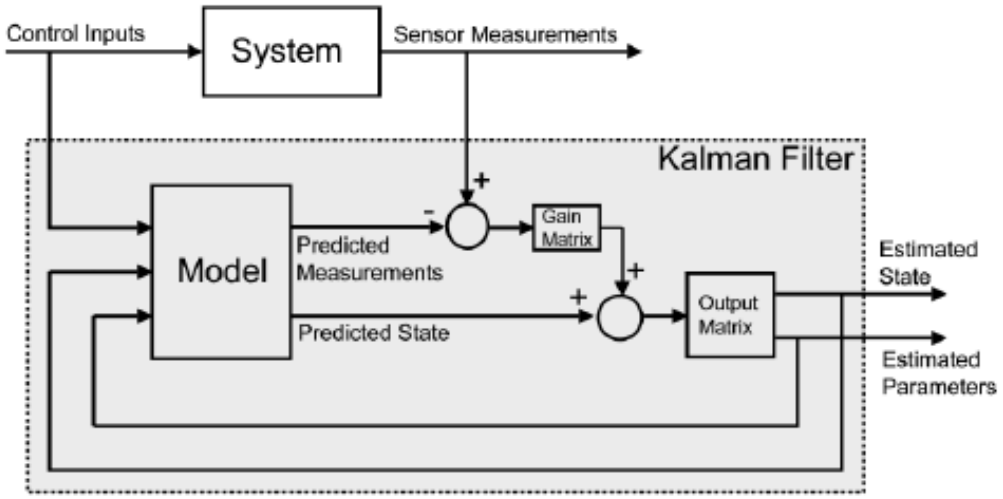

Figure 4.6. A general Block Diagram of the Kalman Filter.

5 KALMAN MODELS

After gaining a solid theoretical and mathematical basis throughout the previous chapters, it is possible now to dive into the analysis of the three proposed models for solving the faced positioning problematics. The models will be explained starting with the most simple and inaccurate model to the most complex and precise one.

5.1 The Constant Velocity Model (CVM)

The first of the three models studied is the Constant Velocity Model (CVM). It is constructed around the assumption that the vehicle moves with a constant velocity, hence there is no acceleration. The physical inconsistencies of the model are obvious, however it is a perfect first approach to the case study and understand how the Kalman Filter performs in this environment. The limitations rely on the fact that the full sample is a route in which the car drives in urban and interurban contexts and the car drives at different velocities in the different moments, hence experimenting acceleration. Nonetheless, it is also true that if the sample was divided in several subsamples in which the velocity variations where smaller, the overall performance would be much better. As a result, the equations that physically described the car motion are the Uniform Rectilinear Motion (URM) ones:

$$x = x_0 + v \cdot t \tag{5.1}$$

$$v = const \tag{5.2}$$

Therefore, from these equations the Kalman Model is constructed- Starting with the State Matrix, its dimensionality is equal to 4. As it is considered that there is no acceleration, there is no reason for the acceleration to be included. Hence, the State Matrix is:

$$\bar{x} = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} \tag{5.3}$$

Where:

- **x** is the UTM “x” coordinate of the current position in meters.
- **y** is the UTM “y” coordinate of the current position in meters.
- $\dot{\mathbf{x}}$ is the velocity in the x-axis in meters per second.
- $\dot{\mathbf{y}}$ is the velocity in the y-axis in meters per second.

Moreover, the Transition Matrix A is squared matrix that is expressed, based on the equations 4.1 and 4.2 and the dimensionality of the State Matrix, as:

$$A = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.4)$$

Where:

- **dt** is the time difference between each point of time that in the GNSS receptor specifications is called “latency”. This is a parameter that is adjusted when configuring the sensor and in the study case was set to 0.15 seconds. However, there were some measures where the latency was inaccurate due to errors in the sensor itself and fluctuated minimally. Hence, the value that **dt** has for this model is *dt = 0.1824 s*.

On the other hand, the bias matrix in the state prediction equation, B, is set up to the following arbitrary values:

$$B = \begin{bmatrix} 0 & 0 & 0.035 & 0 \\ 0 & 0 & 0 & 0.015 \\ 0 & 0 & 10^{-6} & 0 \\ 0 & 0 & 0 & 10^{-6} \end{bmatrix} \quad (5.5)$$

The values that this matrix has are absolutely arbitrary and the aim of them is to manually refine the performance of the prediction. This adjustment must be done trying significant values and studying how the performance evolves. It is also important to note that, however the values are arbitrary, the positions of them in the matrix are not and are consistent with the transition matrix and the dynamic equations that describe the system.

Following with the prediction equations, the Covariance Matrix P has no specific construction beyond the initialisation that will be explained afterwards. However, the Q matrix, the Noise Matrix of the covariance that has a *(4x4)* shape, do have a specific construction. It is computed as:

$$Q = G \cdot G^T \cdot \sigma_v^2 \quad (5.6)$$

Where:

$$G = \begin{bmatrix} 0.5dt^2 \\ 0.5dt^2 \\ dt \\ dt \end{bmatrix} \quad (5.7)$$

And, based on the study by Schuber, R. et al. (2011) named “Empirical Evaluation of vehicular models for ego motion estimation”, the value of σ_v can be set to 8.8m/s^2. With this, the prediction parameters are set to be used for the prediction step.

Moving on to the update step, the first parameter that is going to be commented is the Measurement Observation Matrix H. As it is been said, the sensor is providing the velocity record of the car in each point of time. In fact, the velocity given is the modulus of it and it must be projected in the x and y axes. For this, it is used the track direction with respect to the True North in degrees and, applying trigonometry, the velocity is projected in both axes that is the data with which the model works. It is important to clarify that the True North or Geographic North is the direction towards the fixed point that is called North Pole. Having this clear, the matrix H must be constructed thinking of extracting the desired data from the state matrix, in this case is the velocity in x and y, by computing $H * \bar{x}$. Hence, the Measurement Observation Matrix will be for the CVM:

$$H = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.8}$$

It is important to note that both numerical values are equal to 1 because, in this case, the output of the sensor is exactly what the Kalman Model requires. In the case that the sensor output raw data was not identical to the one used in the model, the values of H could be changed in order to adapt the sensor data to be used in the algorithm. For example, if the output velocity of the sensor was in km/h and the model uses m/s, the matrix H would have to include the conversion factor of the data to set all in the same units.

Moreover, the matrix R that represents the noise of the measurement, is also manually adjusted in order to enhance the performance of the filter:

$$R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{5.9}$$

It is important to understand that these values that appear in the matrix are absolutely arbitrary. Also, it can be highlighted the fact that it is a squared matrix where the dimensionality is the same as the sensor measurement dimensionality.

Finally, the last parameter that is introduced in the algorithm is the sensor measurement matrix Z. For the thesis’ study case it is:

$$Z = \begin{bmatrix} v_{x,sensor} \\ v_{y,sensor} \end{bmatrix} \tag{5.10}$$

Where the values of $v_{x,sensor}$ and $v_{y,sensor}$ are different for each iteration as they are the velocity data for each point of time of the real measurement. These are the reference for correcting the model when it goes through the update step. The predicted velocity is compared with the real one and based on that the algorithm works.

Once the algorithm is set up, the first step is to initialise it. For this purpose, the State Matrix $\bar{x}$ and the Covariance Matrix P are constructed with the real values that the sensor gives for the first point of time. This will be the first and last time throughout the filter in which the real values for x-y position and covariance are used. Once this is done, the algorithm enters the iteration stage and the filter starts its performance. The output of the correction made by the Constant Velocity Model can be seen in the Figure 4.1.:

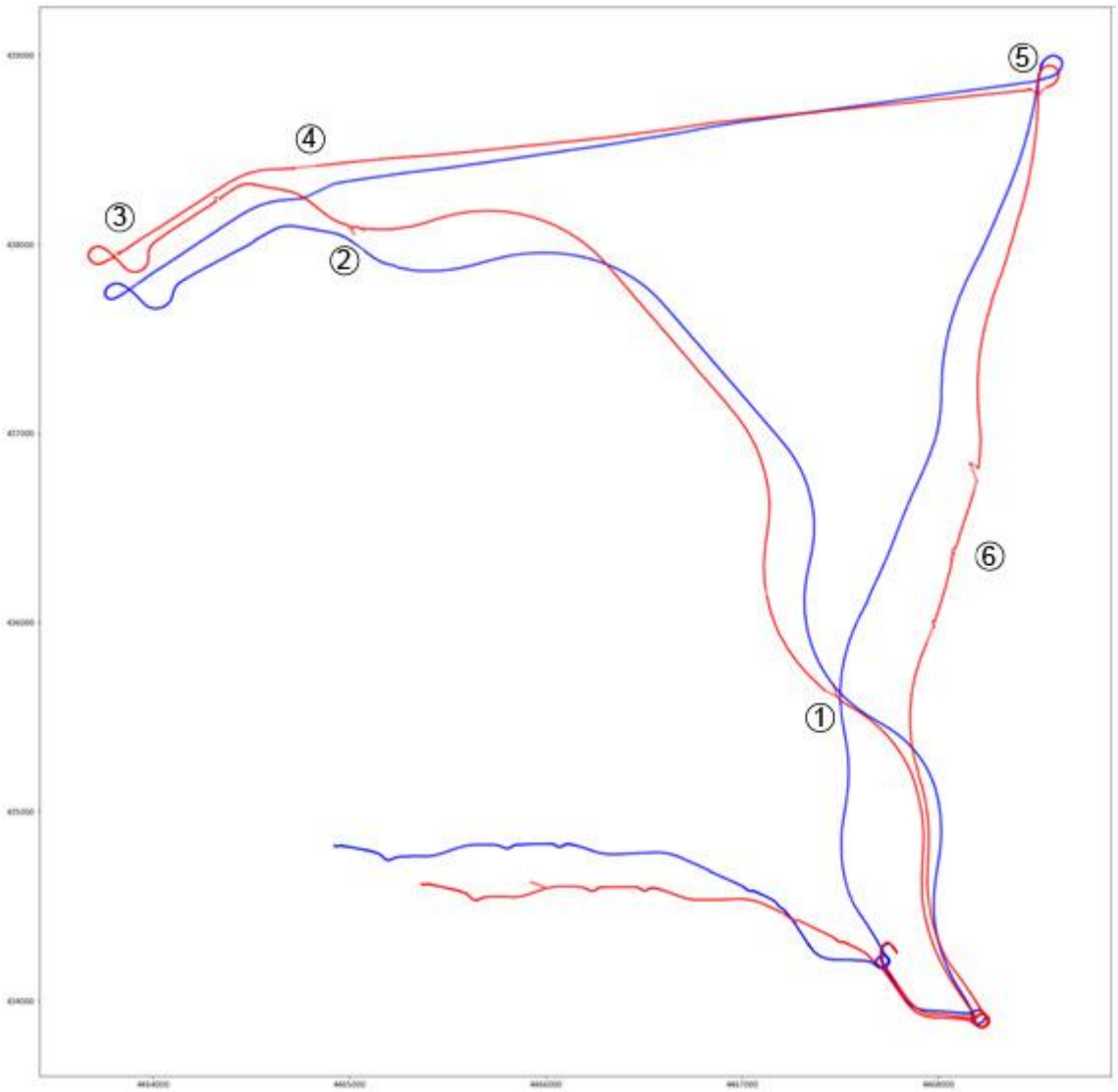

Figure 5.1. In red: the receptor's trajectory. In blue: the CVM corrected trajectory.

Although the physical assumption of a constant velocity was clearly incorrect for analysing the study case, the output performed by the Kalman Constant Velocity Model is considerably good. It draws the same trajectory although proportion is incorrect. This is what could be expected as the velocity remains the same, hence in the sections in which the real velocity is higher than the one established in the model, the projections are made to less meters than they really are. This can be demonstrated by studying equation 4.1.: with less velocity, the final position is lower. There are also big mistakes

at the end of the sample because the error is accumulated throughout the different iterations and produces significant mistakes when the sample is big, as it is the case.

In order to make a deeper analysis of the performance of the Constant Velocity Model, it is also important to pay close attention to the 6 singularities that were selected in the introductory chapter while explaining the nature of the sample that is used in the thesis.

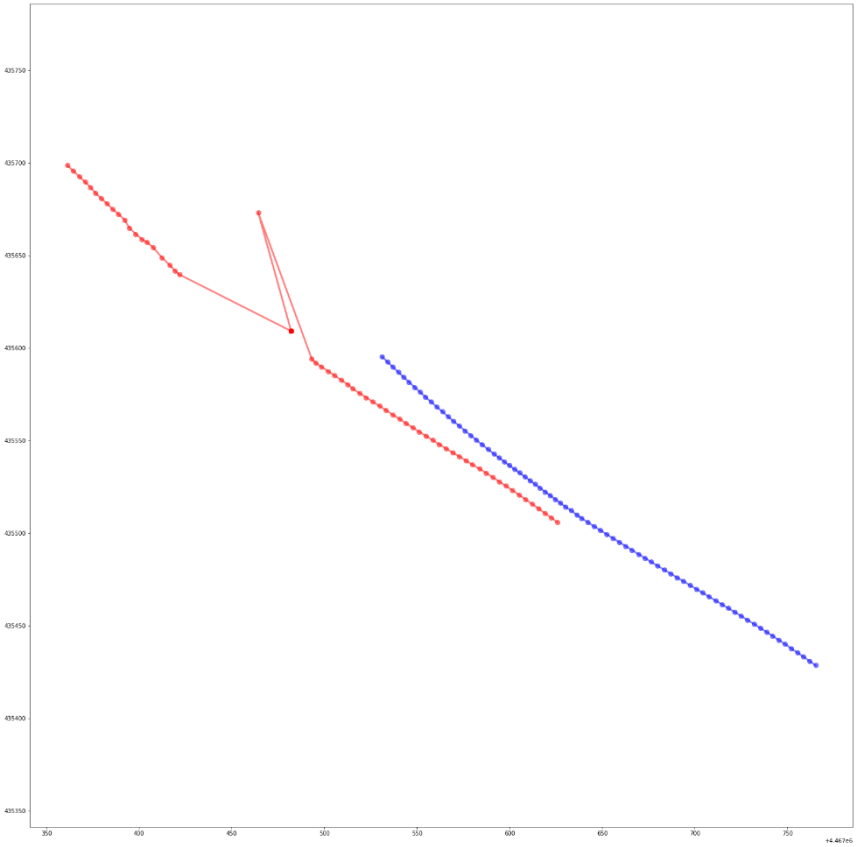

Figure 5.2. Singularity number 1 (in red: the receptor's trajectory. In blue: the CVM corrected trajectory).

Figure 5.3. Singularity number 1 in real life (image extracted from Google Maps).

As it is clearly seen in the Figure 5.2, over the proportional mistakes that are inherent to Constant Velocity Model, the singularity is correctly resolved. While the GNSS receptor outputs a clear positioning error with a point very distant from the trajectory, the Kalman CVM is able to perform a straight line continuing the correct trajectory and solving the singularity correctly. It is also interesting to have a more visual perception of what causes this type of singularities. In this case, Figure 4.3 shows an image from Google Maps in which it is possible to clearly see that the car was driving in a roadway and went below a bridge, losing visibility to the satellites and, hence, losing its connecting for many points of time.

Moving on to the next singularity, it is possible to see how the Model corrects once again the incorrect trajectory of the GNSS receptor. It is true, though, that in this case the CVM performs two jumps that should not occur and that are a result of the inconsistency of the data given by the sensor. The main particularity of this sub-sample is that the wrong measurements are diverse and, more or less, the points progress in the direction of the true trajectory. In the previous singularity, the wrong measurements got stuck in one same point until the connection was recovered and the trajectory was corrected.

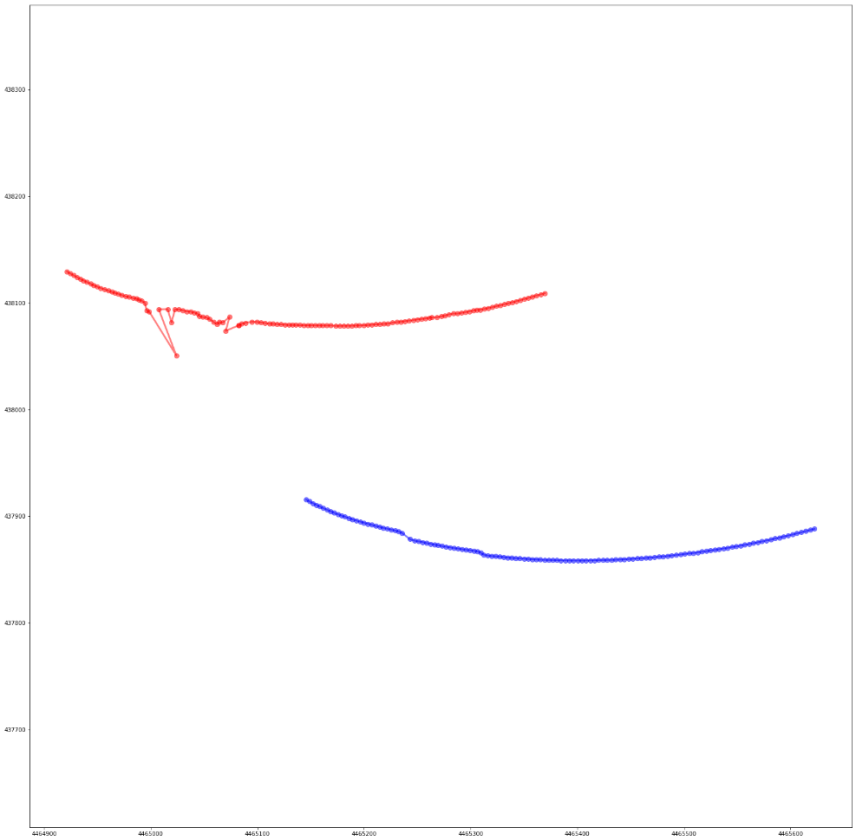

Figure 5.4. Singularity number 2 (in red: the receptor's trajectory. In blue: the CVM corrected trajectory).

It is also an interesting view of how the Kalman Filter is able to work perfectly when describing the required trajectory shape in singularity number 3. At this point of the sample, the car makes several turns in order to make a direction change in the motorway

in which it was currently driving. The shapes of the curves are described perfectly and the major positioning mistake that appears is smoothly solved.

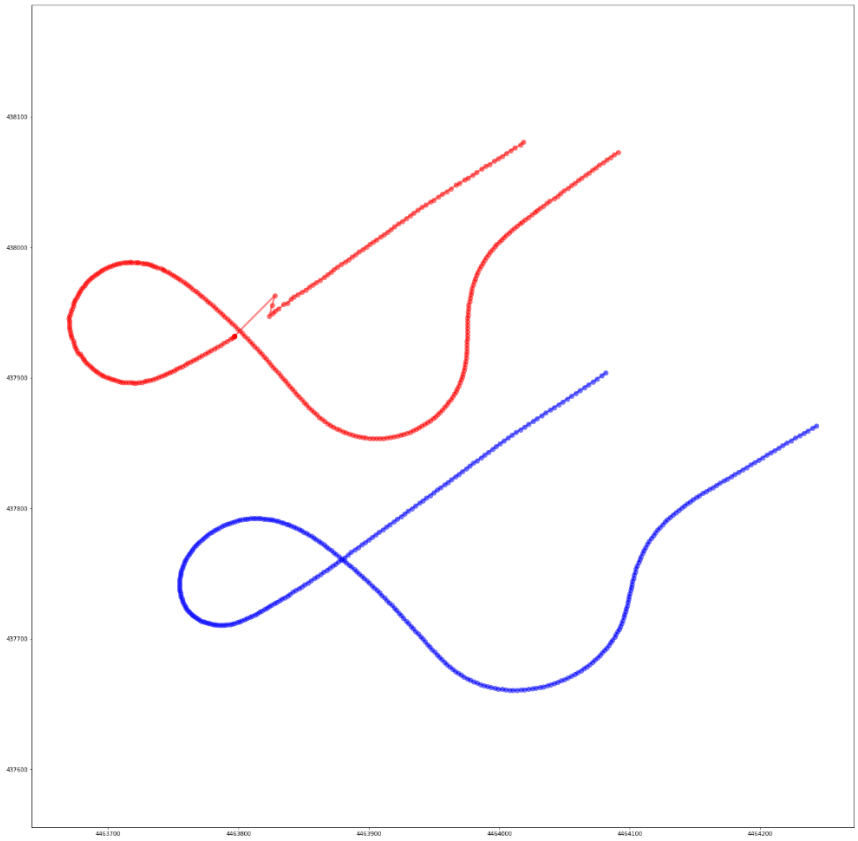

Figure 5.5. Singularity number 3 (in red: the receptor's trajectory. In blue: the CVM corrected trajectory).

The fourth singularity is the most complicated one of all of them as it will be seen during all the Kalman Models although the problematic is once again caused by a bridge.

Figure 5.6. Singularity number 4 in real life (image extracted from Google Maps).

For this singularity, the GNSS receptor gets stuck for 4 seconds in one point covering a total of 27 points of time. However, in this case, the behaviour of the Kalman Filter is not the one that is desired. In order to describe the nature of this singularity, it is very useful to compare the real scenario and how is similar and different from the one of the first singularity, that was correctly solved. As it can be seen in the Google Maps images, in the first case the bridges that cause the signal loss are higher and are more separated one from another. On the other hand, in the case of this fourth critical point, the bridges are much lower and close one from another causing a tunnel effect. Nonetheless, it can be also considered some type of error coming from the sensor itself in the sense that it is impossible that the car takes 4 seconds to drive below the bridges considering the velocity at which a car drives in a motorway in Spain (100-120 km/h) and the distance to be covered from the start to the end of this urban feature.

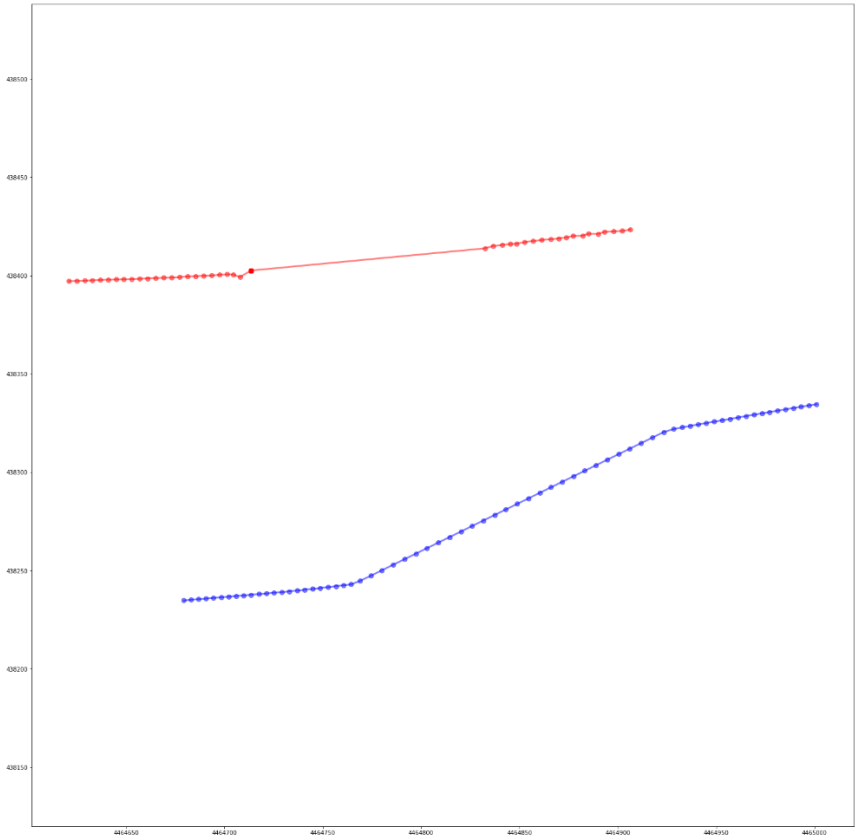

Figure 5.7. Singularity number 4 (in red: the receptor's trajectory. In blue: the CVM corrected trajectory).

The fifth singularity is very similar to third one and it can be seen how the model works perfectly as it solves the errors that appear in the sensor's raw trajectory.

On the other hand, the las singularity to study is a very striking success case of the CVM. In the context of another urban feature that produces signal losses, the filter is able to manage positioning errors that are unacceptable if in an autonomous driving context. With the Kalman Model tis problem is solved and a reliable and useful trajectory is given as a result.

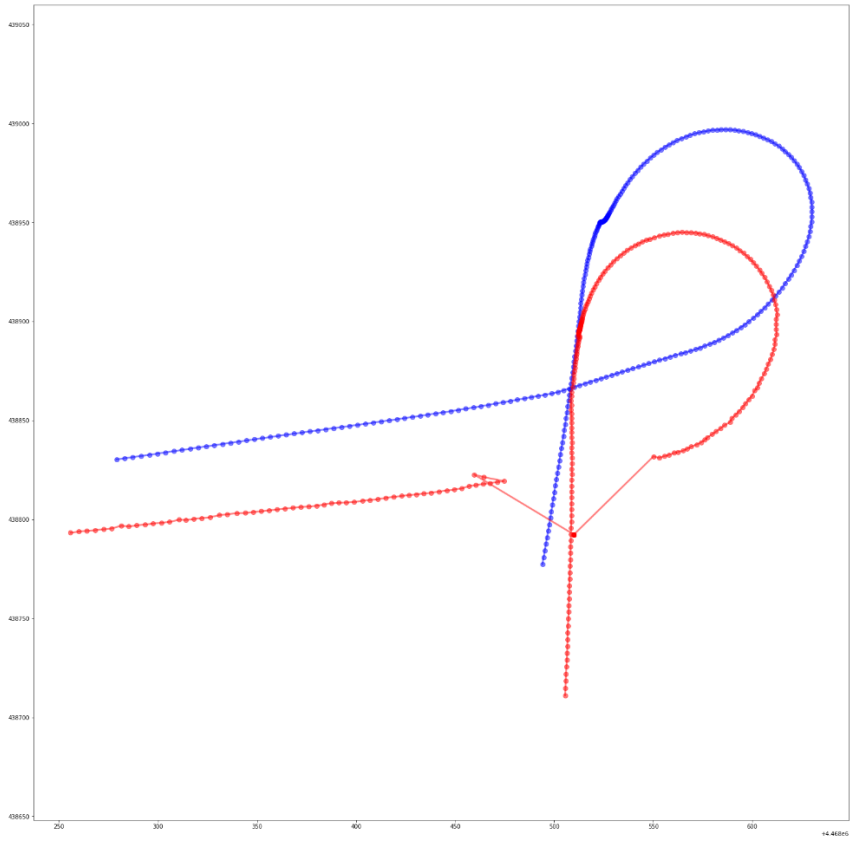

Figure 5.8. Singularity nubmer 5 (in red: the receptor's trajectory. In blue: the CVM corrected trajectory).

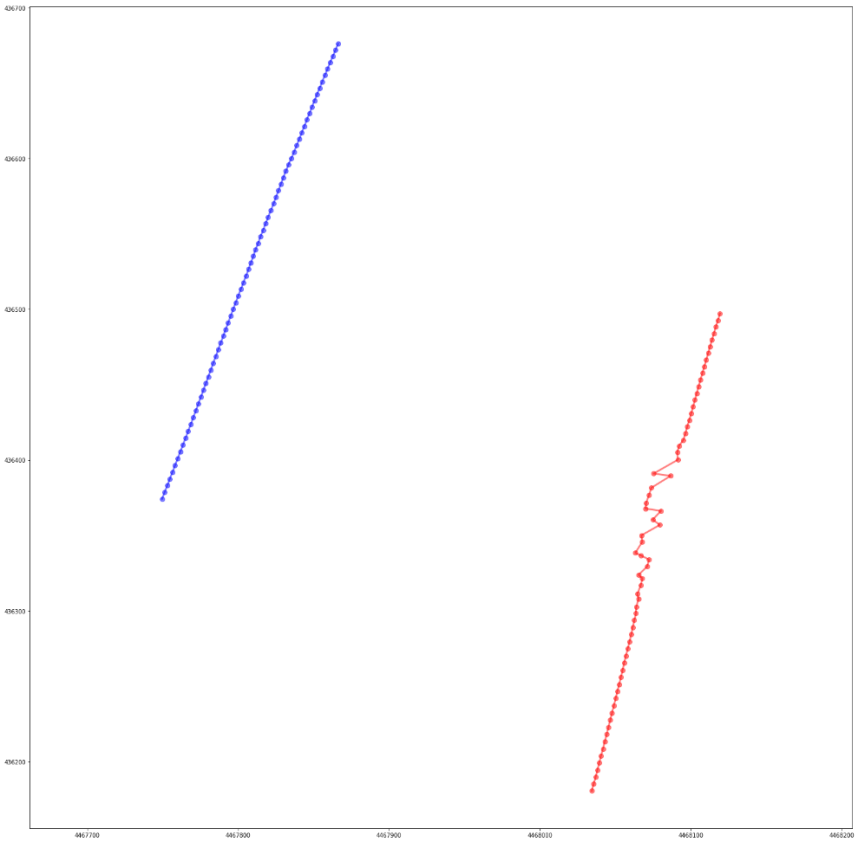

Figure 5.9. Singularity number 6 (in red: the receptor's trajectory. In blue: the CVM corrected trajectory).

5.2 The Constant Acceleration Model (CAM)

The second model is the Constant Acceleration Model. As it can be inferred from its name, the dynamic behaviour of the car is modelled considering that the acceleration is constant whereas the velocity does change. This model reproduces much better the system and performs a more reliable prediction of the trajectory. However, it is also true that the acceleration may not be the same for the whole sample but as velocity changes are considered, the model's performance is sufficiently good. Hence, the dynamical model is based on a Uniformly Accelerated Rectilinear Motion (URMA):

$$x = x_0 + v_0 t + \frac{1}{2} a t^2 \tag{5.11}$$

$$v = v_0 + at \tag{5.12}$$

Therefore, the State Matrix has a dimensionality of 6, as acceleration in both x and y axes is considered:

$$\bar{x} = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \end{bmatrix} \tag{5.13}$$

Where:

- $\ddot{\mathbf{x}}$ denotes the acceleration in the x axis in m/s^2.
- $\ddot{\mathbf{y}}$ denotes the acceleration in the x axis in m/s^2.

As the State Matrix and the system equations have changed, the Transition Matrix also performs changes in order to comply with the specific requirements:

$$A = \begin{bmatrix} 1 & 0 & dt & 0 & 0.5dt^2 & 0 \\ 0 & 1 & 0 & dt & 0 & 0.5dt^2 \\ 0 & 0 & 1 & 0 & dt & 0 \\ 0 & 0 & 0 & 1 & 0 & dt \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.14}$$

Where:

- **dt** denotes the latency of between each point of time that, as in the Constant Velocity Model, is set to 0.15 seconds but due to errors in the sensor it is slightly unstable during the sample and, hence, it is taken the average value: *dt = 0.15s*.

Moving on, the bias matrix for the state prediction step is set to:

$$B = \begin{bmatrix} 0.0015 & 0 & 16500 & 0 & 0 & 0 \\ 0 & 0.045 & 0 & 12500 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot 10^{-6} \tag{5.15}$$

It is important to remember that the values of this matrix are set arbitrarily and after a manual adjustment procedure.

On the other hand, the matrix P has the only difference in the fact that its shape is *6x6* to be consistent with the model dimensionality. Moreover, the noise matrix Q is constructed following the same equation 4.6 and making the same assumption for the value of σ_v (equal to 8.8 m/s^2):

$$G = \begin{bmatrix} 0.5dt^2 \\ 0.5dt^2 \\ dt \\ dt \\ 1 \\ 1 \end{bmatrix} \tag{5.16}$$

$$Q = G \cdot G^T \cdot \sigma_v^2 \tag{5.17}$$

The Observation Matrix H has some changes and it is constructed as:

$$H = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{5.18}$$

As the sensor information is the same, the velocity in the x and the y axes, the Observation Matrix must be constructed to extract those same values but also satisfy the new dimensionality of the system.

Moreover, the Noise Measurement Matrix R is set to:

$$R = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix} \tag{5.19}$$

Finally, it is important to remember that the measurement matrix Z stays the same for the three models as:

$$Z = \begin{bmatrix} v_{x,sensor} \\ v_{y,sensor} \end{bmatrix} \tag{5.20}$$

Having all the parameters set for this model, it is possible to discuss the results of the model. In Figure 4.8, it is shown the full trajectory and the Constant Acceleration Model correction.

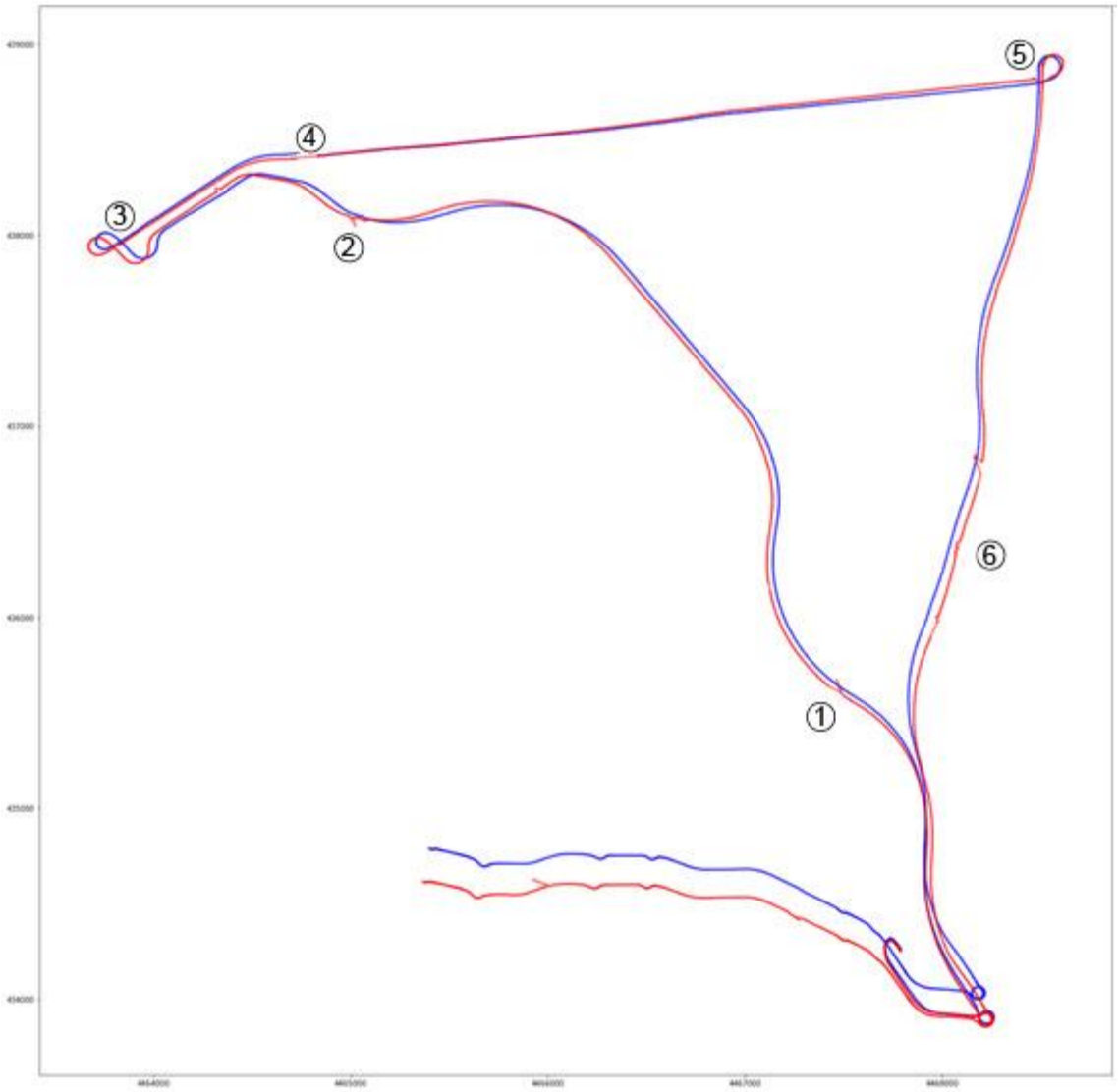

Figure 5.10. In red: the receptor's trajectory. In blue: the CAM correction

As it can be seen, the performance of the filter is much better for this case. During the whole trajectory follows correctly the trajectory described by the car and corrects the positioning mistakes that occur eventually. However, at the end of the sample it true that the model diverges slightly. It is important to understand that this is something that happens inevitably for so long datasets as the error accumulation eventually is enough high to create inconsistencies. In order to make a deeper analysis on the performance of the filter, it is better to highlight and explain the six singularities that have been selected to demonstrate how the filter overcomes GNSS inconsistencies.

Firstly, the singularity number 1 is absolutely solved as it can be seen in Figure 5.9.:

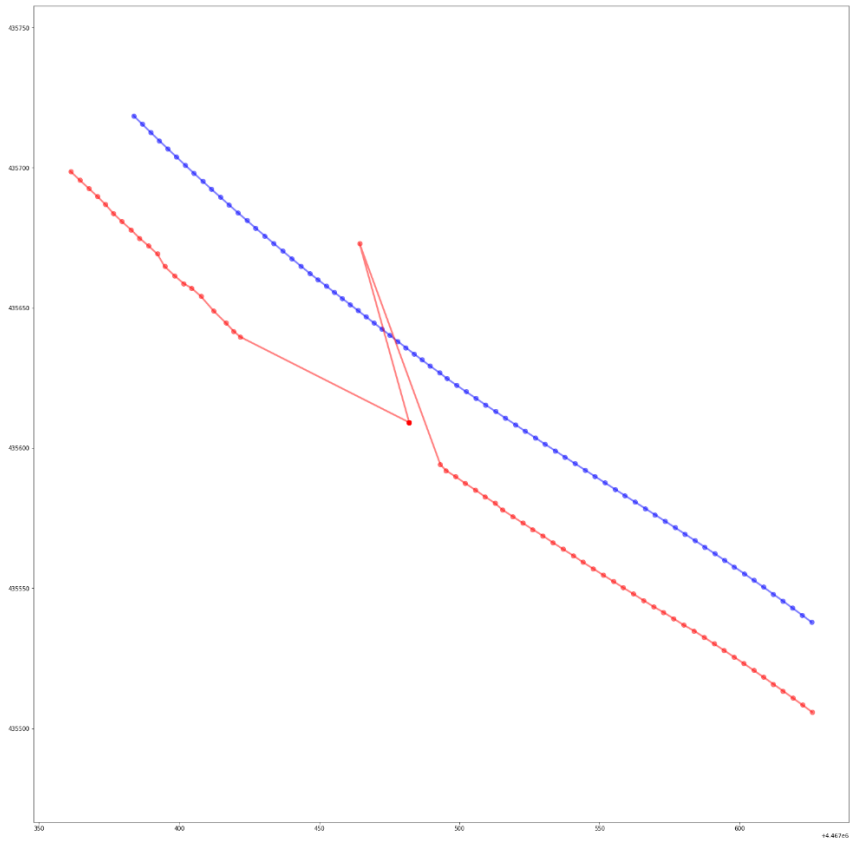

Figure 5.11. Singularity number 1 (in red: the receptor trajectory. In blue: the CAM corrected trajectory).

Although the GNSS sensor gets slightly stuck at some point of the trajectory, the Kalman Filter is capable of overcoming this issue and predict a correct path.

On the other hand, for singularity number 2 (Figure 4.10) the jumps that appeared in the Constant Velocity Model still happen but smoother. It is true, though, that the data coming from the sensor is very inconsistent in comparison to the Kalman output and the trajectory that is predicted by the algorithm is much more consistent and robust. In addition to this, it is also the moment to pose a question over the absolute correctness of the velocity measurements of the receptor. Although it is true that the reliability of it is very high because in any other case the model would not respond so good, it is normal that some mistakes are made by the sensor when computing the values of velocity.

Moreover, the singularity number 3 is perfectly resolved and the errors that appear in the path described by the data that the sensor outputs, disappear. Also, the different curves that the care describes are perfectly reproduced in the model and proportions remain exactly the same throughout the sub-sample. In this case there are no jumps in the predicted trajectory because the errors in the sensor data come from an absolute loss of signal for a short period of time, whereas in the previous singularity they occurred because of a worse signal sustained in a longer period of time but the signal was never completely lost. It is interesting to note that the model overcomes better the scenario of the third singularity than the scenario of the second one.

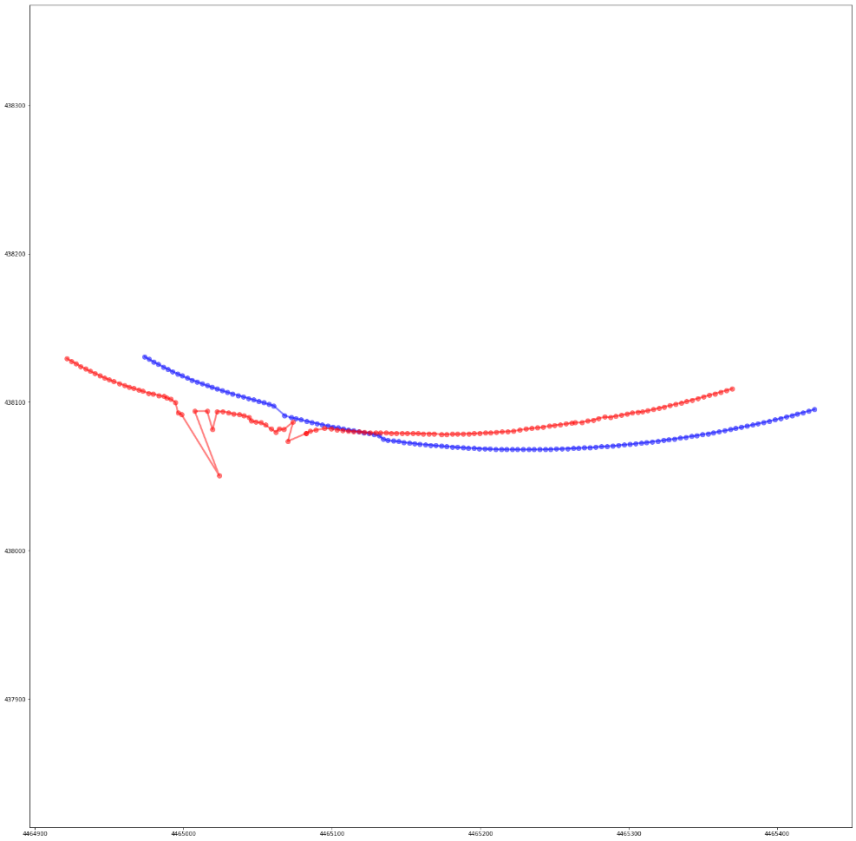

Figure 5.12. Singularity number 2 (in red: the receptor trajectory. In blue: the CAM corrected trajectory).

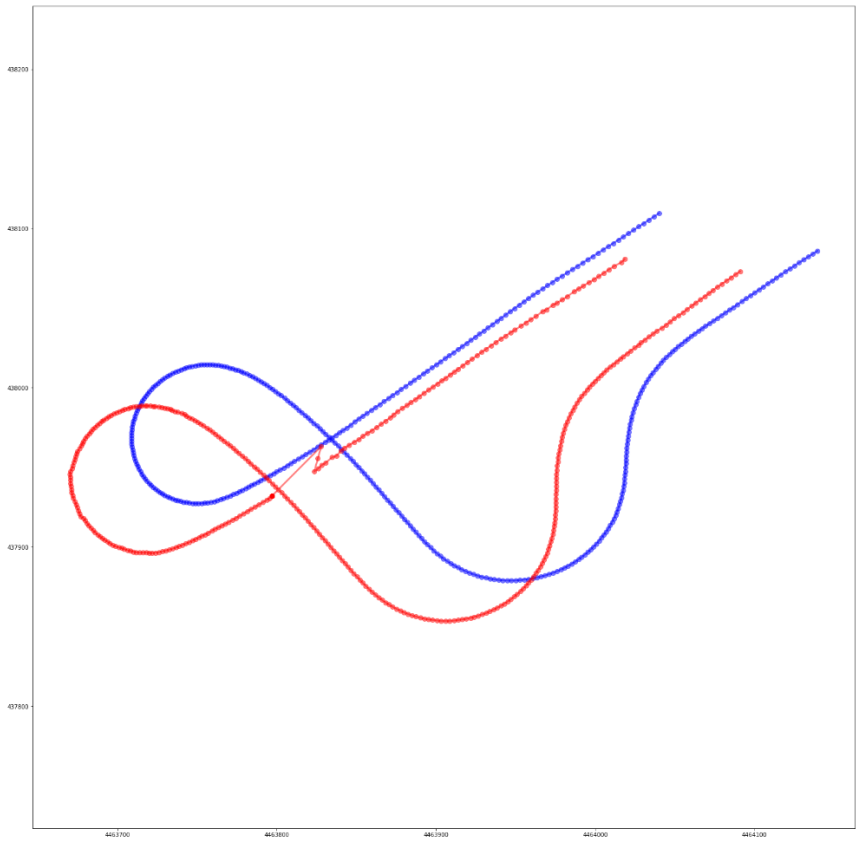

Figure 5.13. Singularity number 3 (in red: the receptor trajectory. In blue: the CAM corrected trajectory).

The next singularity, the fourth one, has been the most problematic of all six singularities. The main issue that appears in this case is that the sensor gets stuck for 4 seconds as it is been said during the analysis of the Constant Velocity Model. It is important to say that not only the position, but the velocity measurements, give repeated data during this period. This time is so big that there it is demonstrable that the sensor itself has some inconsistencies that are limiting the performance of the Kalman Filter and the different models applied. As it is going to be seen in the next section, even the UKF makes errors when computing this part of the sample. It is also important to note how the trajectory from the repeated point and the previous point is the trajectory that the filter reproduces and causes the error in it. In order to overcome this inconsistency, the decision made was to reset the filter when the sensor was capable of output a reliable position. As this singularity is produced by errors from the sensor performance that should not happen, the best way to surmount this juncture is by resetting the filter with the condition of having repeated the same exact point (exactly the same datum of x-position, y-position and velocity) more than 21 times. Once solved this issue, the filter goes on performing the corrected trajectory for the given sample.

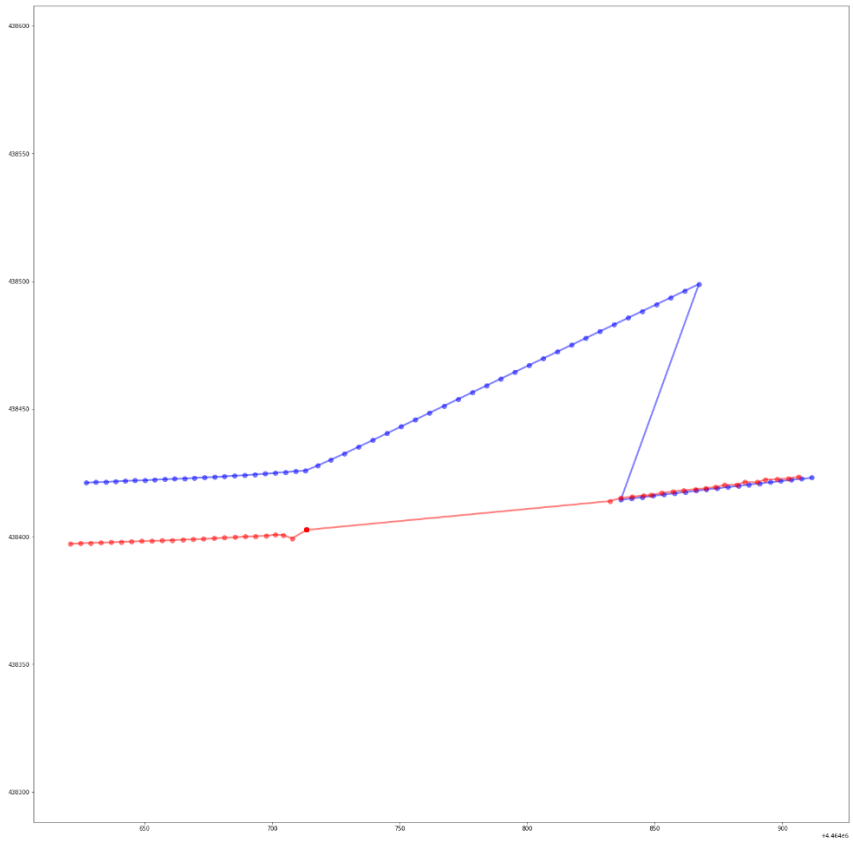

Figure 5.14. Singularity number 4 (in red: the receptor trajectory. In blue: the CAM corrected trajectory).

In the case of the fifth singularity there is a point that gets stuck for more than fifteen points of time, however, the filter is capable of overcoming the inconsistency with no major problems. In any case, driving below a bridge does not justify errors this big and it is important to highlight the difficulties that the Model overcomes for giving such a precise correction.

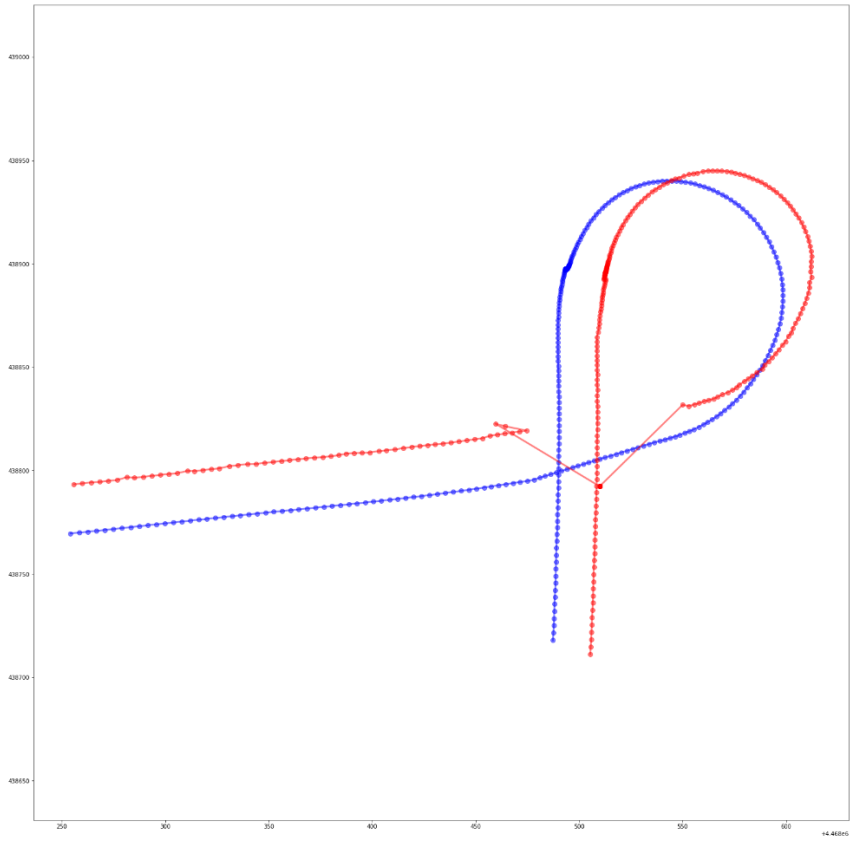

Figure 5.15. Singularity number 5 (in red: the receptor trajectory. In blue: the CAM corrected trajectory).

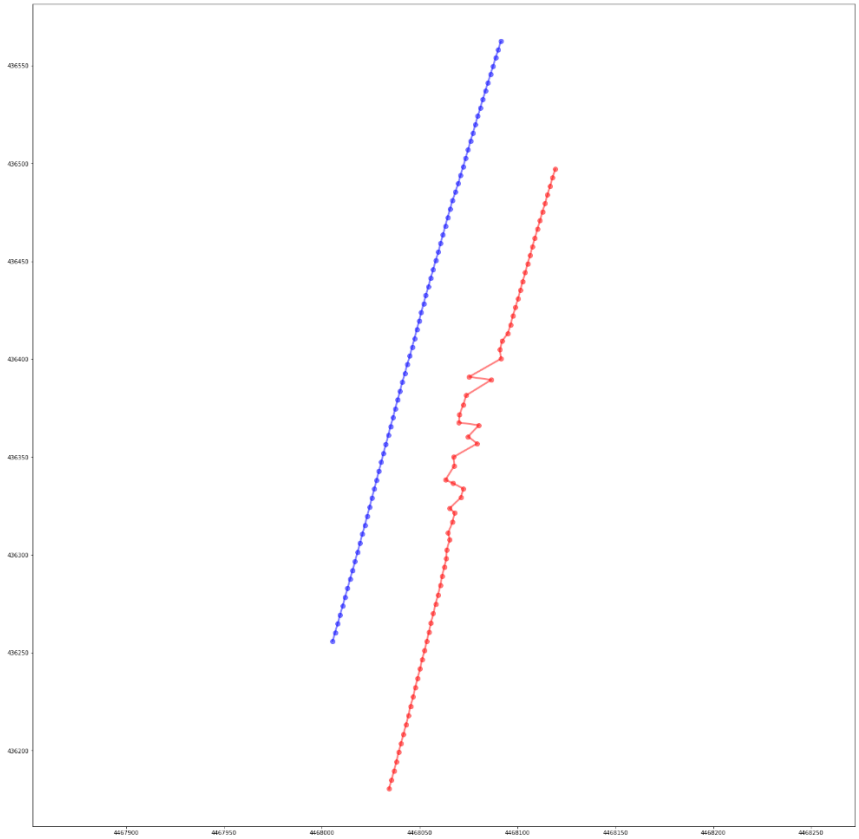

Figure 5.16. Singularity number 6 (in red: the receptor trajectory. In blue: the CAM corrected trajectory).

Finally, the last singularity that gives a deeper perspective on the performance of the Constant Acceleration Model shows that the filter has no problem in overcoming the situation. It is interesting to note that this singularity is very similar to the second one, but, in this case, there are no jumps in the predicted model. This is because in the sixth singularity a line is drawn by the car and in the previous case it was a curve. Hence, the Kalman Filter finds more complicated these situations when happen in a turn of the car.

5.3 The Unscented Kalman Filter Model (UKFM)

The last model that is implemented in this thesis is the Unscented Kalman Filter considering constant the acceleration and making all the physical assumptions that constructed the Constant Acceleration Model. The main difference resides in the fact that the Unscented Kalman Filter is capable of being much more precise than the normal Kalman Filter and is able to overcome non-linearities much better. Hence, most of the parameters remain the same except for the bias matrix B, that it is not used for this model, and the Noise Measurement Matrix whose values change:

$$\bar{x} = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \end{bmatrix} \tag{5.21}$$

$$A = \begin{bmatrix} 1 & 0 & dt & 0 & 0.5dt^2 & 0 \\ 0 & 1 & 0 & dt & 0 & 0.5dt^2 \\ 0 & 0 & 1 & 0 & dt & 0 \\ 0 & 0 & 0 & 1 & 0 & dt \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.22}$$

$$G = \begin{bmatrix} 0.5dt^2 \\ 0.5dt^2 \\ dt \\ dt \\ 1 \\ 1 \end{bmatrix} \tag{5.23}$$

$$Q = G \cdot G^T \cdot \sigma_v^2 \tag{5.24}$$

$$H = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{5.25}$$

$$R = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix} \tag{5.26}$$

Initialisation is also done in the same way as the previous ones and there is no need for further explanation. Therefore, it is possible to go directly to the analysis of the model performance. In the Figure 4.15, it is showed both the original trajectory and the corrected one through the Unscented Kalman Filter Model. It is important to highlight that, although scaling parameters must be adjusted, at a first glimpse it is possible to notice that the accuracy has improved. In any case, in the following pages all six singularities will be analysed.

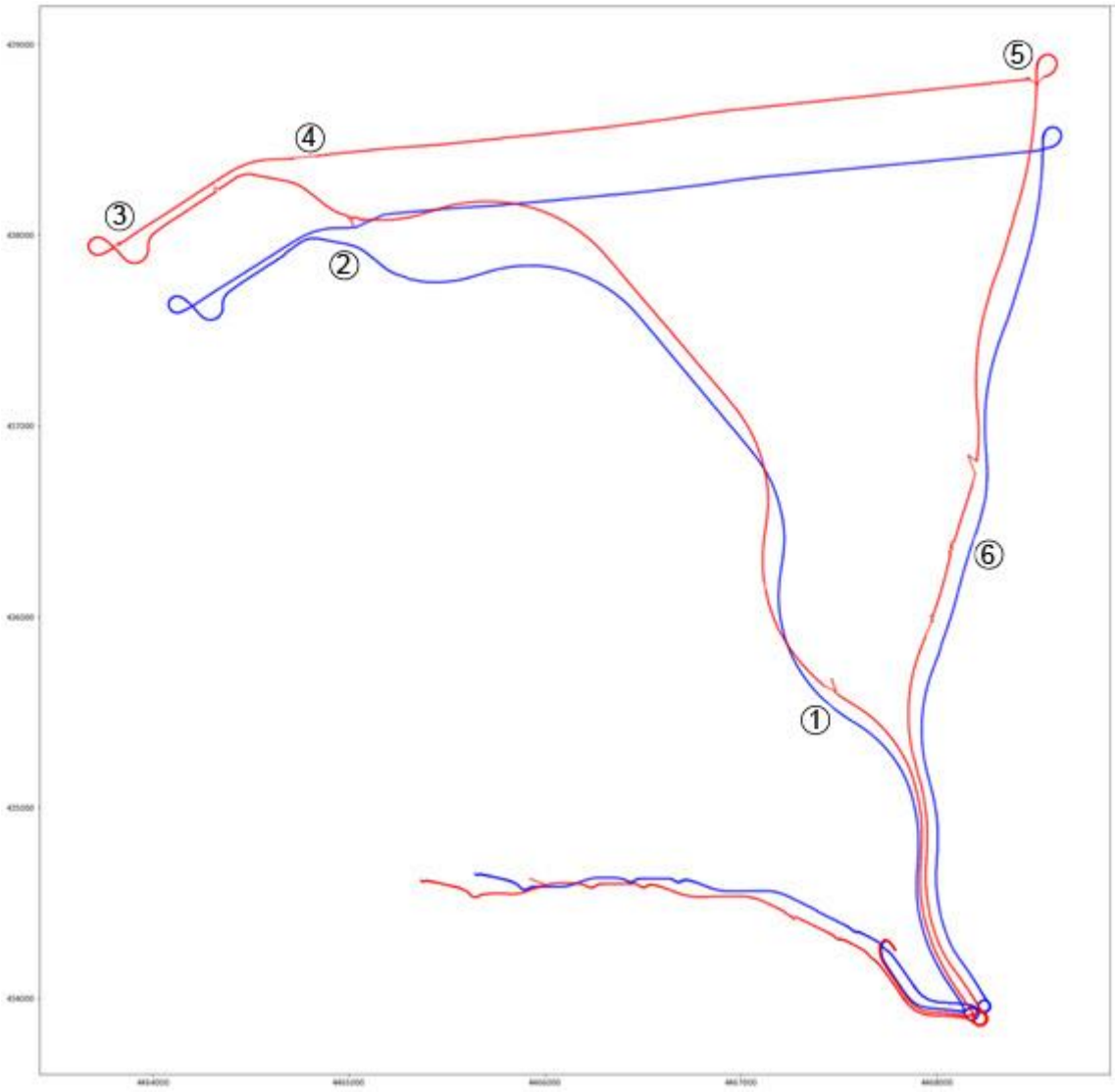

Figure 5.17. In red: the receptor's trajectory. In blue: the UFKM correction.

For the first singularity, it is possible to see that the Kalman Filter performs no errors in the trajectory correction.

The second study point shows a similar problematic to the previous models. Those small jumps appear again although the overall trajectory performs a good correction. As this is something that has appeared in the three models, it is possible to determine that is has to do with inconsistencies in the sensor.

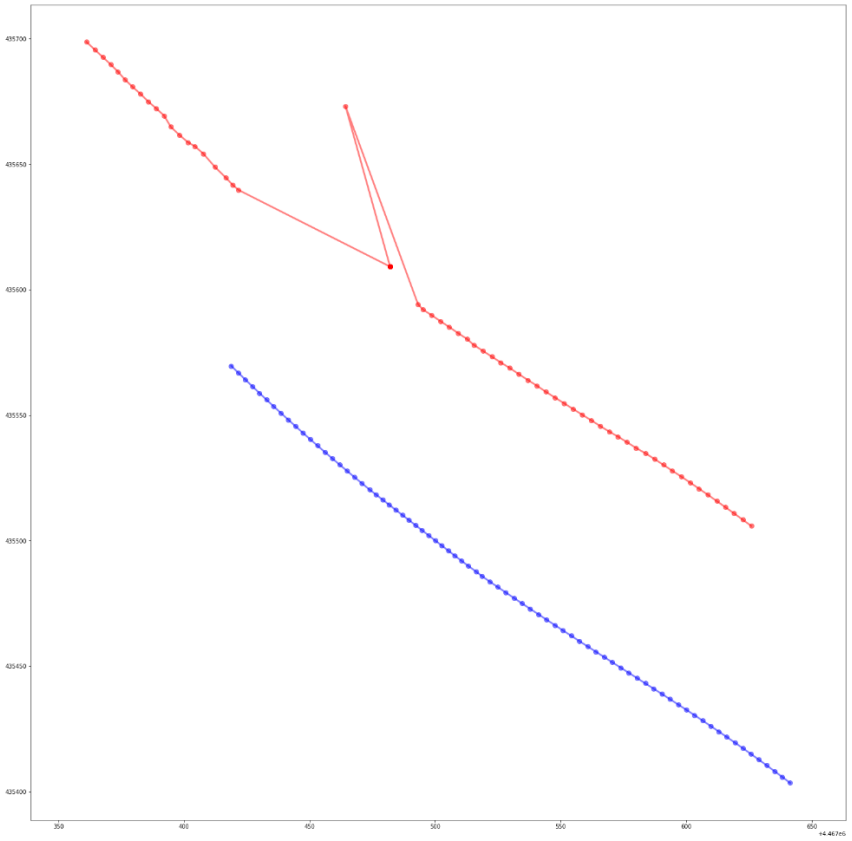

Figure 5.18. Singularity number 1 (in red: the receptor trajectory. In blue: the UKFM corrected trajectory).

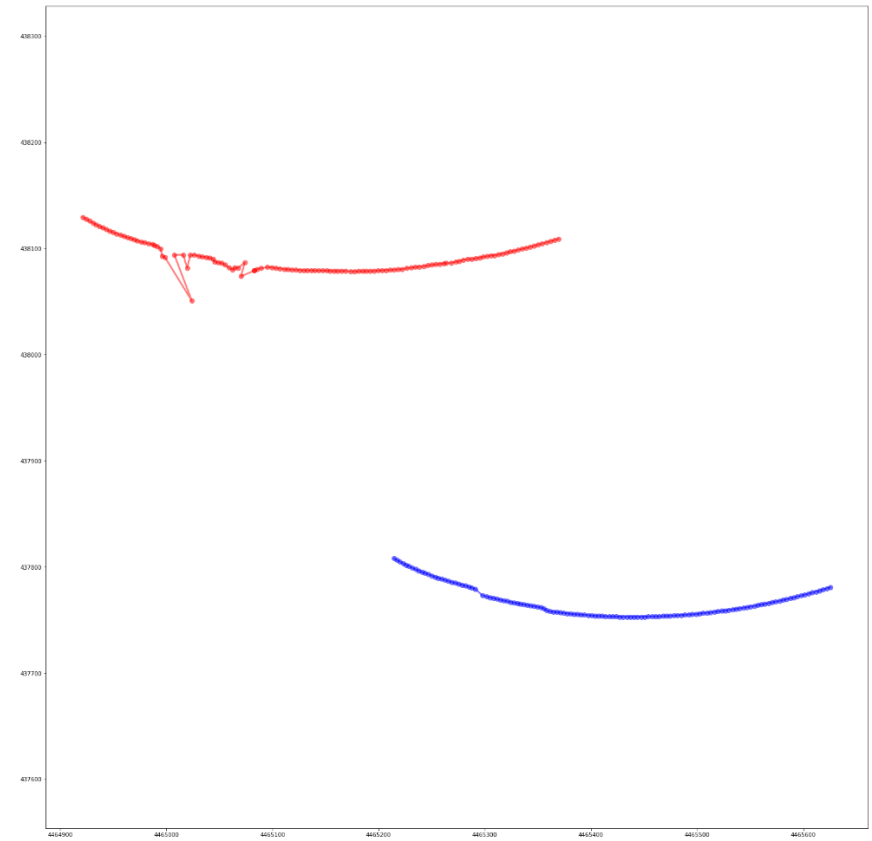

Figure 5.19. Singularity number 2 (in red: the receptor trajectory. In blue: the UKFM corrected trajectory).

The third singularity is also corrected perfectly and the curves are perfectly reporduced in the model.

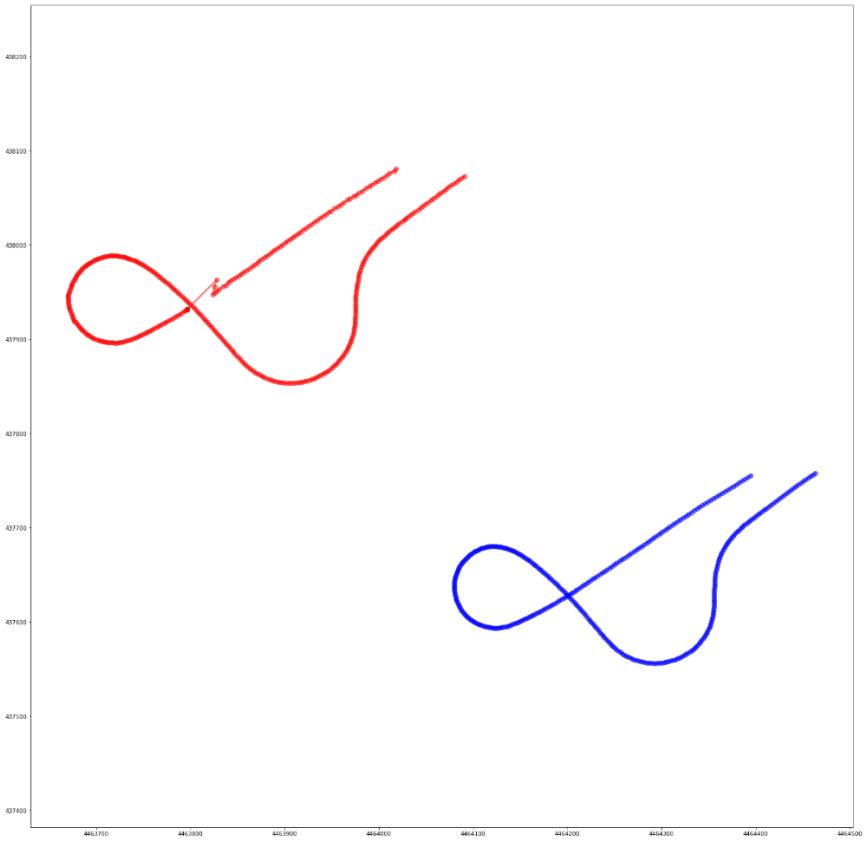

Figure 5.20. Singularity number 3 (in red: the receptor trajectory. In blue: the UKFM corrected trajectory).

The fourth singularity has the same problem as in the previous models. However, in this case, the filter reset has not been introduced as first it is needed to adjust the scaling factors for the algorithm. Once they are perfectly adjusted, the same solution will be deployed.

On the other hand, the fifth point of study is the best prove of the precision improvement of the Unscented Kalman Filter Model. In the previous models, it was possible to see small imprecisions when the curve was described. In the case of this model, the shape is perfect. In addition to this, it is also possible to see that the main problem that this singularity has is also better resolved. There is a point that is wrong and outputs the same wrong datum for several points of time and, in this case, the performance of the prediction is better as the line described is slightly more accurate in comparison to the previous models.

Finally, the last singularity is resolved perfectly and there are no problems with the predicted model.

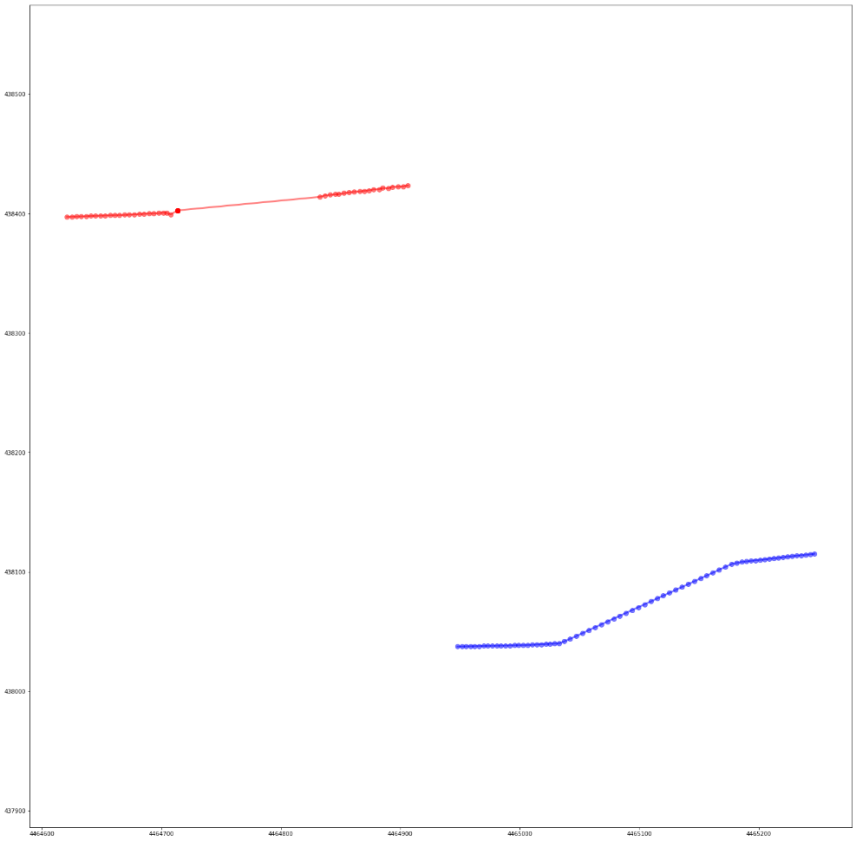

Figure 5.21. Singularity number 4 (in red: the receptor trajectory. In blue: the UKFM corrected trajectory).

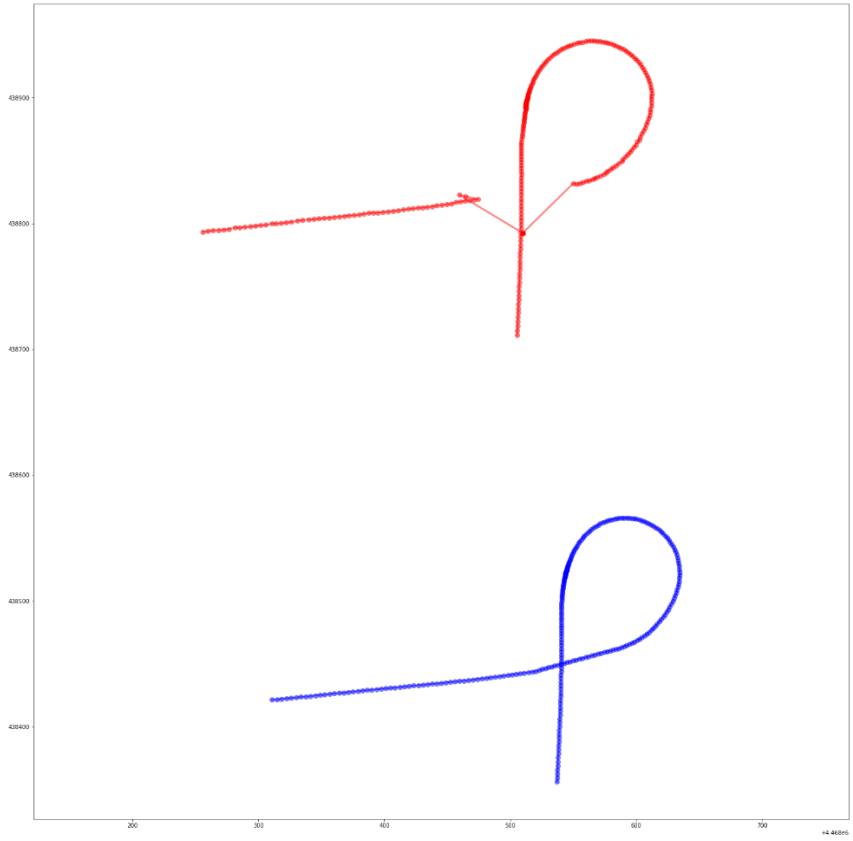

Figure 5.22. Singularity number 5 (in red: the receptor trajectory. In blue: the UKFM corrected trajectory).

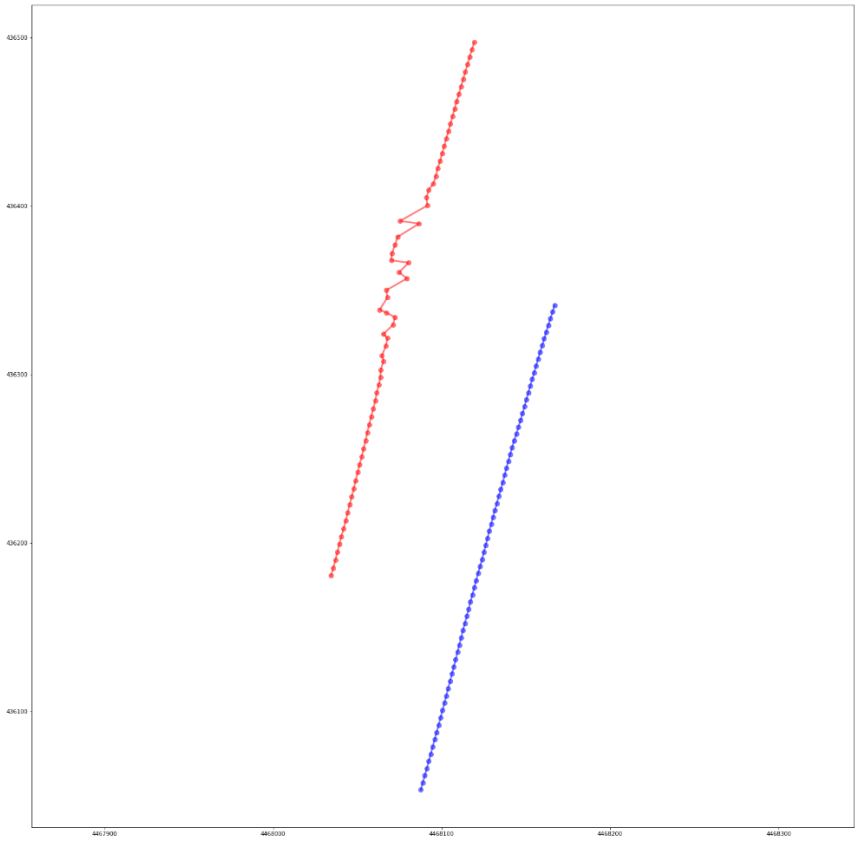

Figure 5.23. Singularity number 6 (in red: the receptor trajectory. In blue: the UKFM corrected trajectory).

6 THE RESULTS

After making a deep analysis of the three models separately, it is also important to make an overall analysis on the performance of the Kalman Filter and the differences and improvements that the different models have.

One of the most visual things that can be seen at a first glimpse is how the error is increased at the end of the sample. This is because through the Kaman Filter it happens an error accumulation that is, perhaps, one of the main downsides of the algorithm. In fact, this is the reason why the Extended Kalman Filter was discarded in order to obtain a better predictive model as the EKF has a very high error accumulation rate. Also, as it is expected, the Unscented Kalman Filter Model is the one that has the least error in the final part of the sample.

It is also possible to see how the precision in the prediction increases from the Constant Velocity Model to the Unscented Kalman Filter Model. However, the performance of the Constant Acceleration Model is considerably good as it has been explained before. The precision of the filter is enough high and the main point in which overcomes the Unscented Kalman Filter Model is the computational load.

In first place, it is important to note that the three systems are very quick in giving predictions mainly because Python is a very optimal programming language that performs very fast this kind of mathematical computations. However, it is true that the Unscented Kalman Filter is computationally heavier that the normal Kalman Filter because of all the extra computations that are made involving the unscented transform, the sigma points and weights. In addition to this, the UKF implementation is based on the work of Jouni Hartikainen, Arno Solin and Simo Särkkä from Aalto University in Finland. They develop an implementation in MATLAB and, hence, an API is needed to communicate from Python with MALTAB and being able to us their work. This process also introduces slowness to the model. On the other hand, the CVM and CAM were fully implemented in Python without using any external work.

Secondly, it is possible to see that the Constant Acceleration Model is the one that is better fitted throughout the whole trajectory. The Constant Velocity Model is fitted very poorly and it presents distortion in the shape proportion that is especially visual in the singularity number 5. On the other hand, the Unscented Kalman Filter Model is not scaled as there are no scaling factors applied so that the model fits correctly to the proportions of the real trajectory. However, it is expected to perform perfectly when dimensioned correctly to the reference proportions.

Moreover, it is possible to determine that singularity number 4 has caused problems to the three models, while others vary in each one. The main issue here is that the signal loss is prolonged for a long period of time and, hence, the singularity is especially severe. It should also be taken into consideration that the previous points of the singularity itself (when 27 exact measures are repeated) are very inconsistent and also have a very important error that makes it even harder to the model to make a correct adjustment of the trajectory for this case. However, restarting the filter makes it possible to overcome this singularity and the algorithm performs correctly for the rest of the sample.

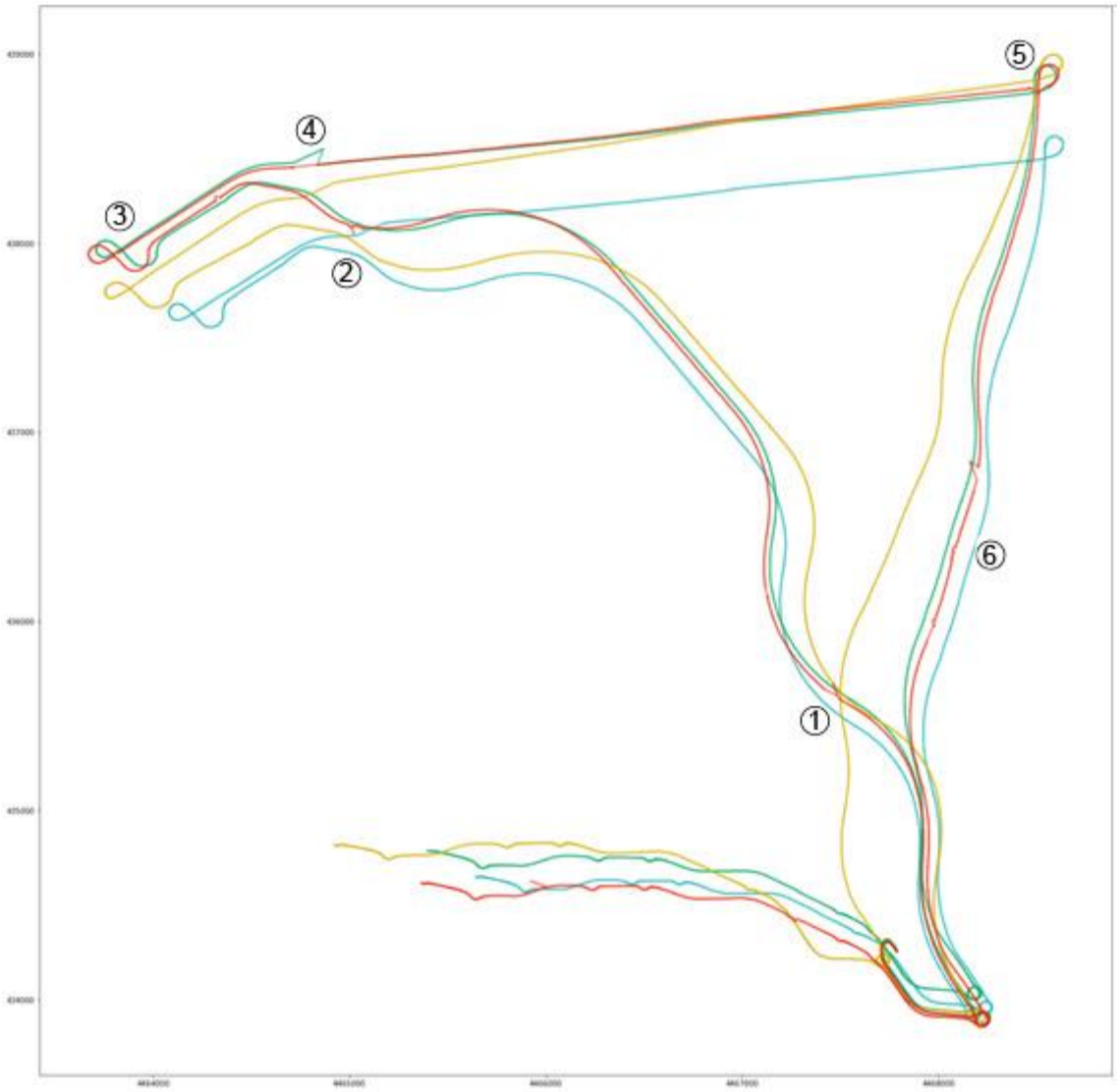

Figure 6.1. All four samples are shown superposed in the same graph. In red: Original trajectory. In gold: Constant Velocity Model. In green: Constant Acceleration Model. In blue: Unscented Kalman Filter Model.

7 REGULATORY FRAMEWORK

The regulatory framework of this thesis depends on three main parts: autonomous driving, GPS legislation and data protection laws.

The first of them, is the most controversial and underdeveloped of all of them. As the autonomous driving technology is quite recent, legislators and governments have not reacted yet to it. Most advances made on this topic that affect Spain are being done at the European Union level. However, there in Spain there is the "Instrucción 15/V-113" from the "Dirección General de Tráfico" that is the competent institution in the country for traffic regulation and management. This document sets a definition of what an autonomous vehicle is as well as the difference between an autonomous mode and a conventional mode. It also regulates, and this is the main purpose of the norm, under what circumstances a licence for autonomous vehicles experimentation can be given. In Spain, this is the only rule that exists in the field of autonomous driving and is limited to experiments made by companies, universities or research centres under the expedition of a specific license whose terms are exposed in this document. At the EU level, the most relevant movement is a resolution from 15 January 2019 in which the European Parliament sets 81 points to be followed by the institutions in order to advance in the development of the regulatory framework that permits the evolution of this technology. The main regulatory institution that is working on this issue is UNECE (United Nations Economic Commission for Europe) that is developing frameworks since 2014. However, Germany already legitimated autonomous driving in 2017 under the condition of the existence of a driver that is ready in any moment to take over the control of the car. Therefore, in general in Europe there are not laws but only norms or propositions that still have to be developed. On the other hand, USA is a step forward in autonomous driving legislation as already in 2017 there was a law approved in the Congress that regulated the sector. The three main points of this law are:

1. It is compulsory that a human driver is supervising all the time and is ready to take over the control of the car.
2. This driver must have done a specific course for this purpose.
3. Vehicles must have the capacity of driving in "safe mode" in the case there is a software problem. In addition to this, if the vehicle determines that it cannot continue driving, it will immediately stop.

Moreover, the most important progress is been made at a state level where legislation is being approved on this matter and the state of Arizona is on the lead. In fact, Waymo's taxi service "Waymo One" operates in the Phoenix, capital of Arizona as well as other companies such as Uber.

In second place, the legislation for the GNSS systems is very long and treats very different situations from fishing to employee localisation during work time. However, there is no specific legislation at the Spanish level for geo-positioning of vehicles. The legislation at the European level is mainly about the GNSS systems themselves but not including the specific case that is studied in this thesis. The most important regulation is the "Regulation (EU) No 1285/2013 of the European Parliament and the Council of 11

December 2013 on the implementation and exploitation of European satellite navigation systems and repeating Council Regulation (EC) No 876/2002 and Regulation (EC) No 683/2008 of the European Parliament and of the Council". The main purpose of this regulation is to set a framework for the European Global Geostationary Navigation Overlay Service and its fit in the context of the European Horizon 2020 programme.

Finally, it is important to note that there is also a data-related legislation that affects the thesis as it works inherently with data. The main regulation on this matter is the General Data Protection Regulation (GDPR) that was proposed in 2012, made in 2016 and approved in May 2018. This legislation addresses the transfer of personal data inside and outside the EU and EEA areas as long as a European individual, company or institution is concerned.

8 SOCIOECONOMIC REPORT

This section is divided in two main parts: the first one is a budget estimation for the development of the technology proposed in the thesis and the second one is a socioeconomic study of the impact of it.

8.1 Budget

The estimated budget for developing the project includes three main variables: documentation, Research and Development

BUDGET

Activity	Hours	€/hour	Cost (€)
Documentation	60	25	1 500
R & D	160	50	8 500
FlexG2 Novatel	-	-	9 000
Fuel	-	-	35
Software	-	-	0
TOTAL			19 035

Table 8.1. Shows the Budget for the whole project.

8.2 Social and Economic impact

The impact of the developed technology can be very high mainly in the western developed countries as well as other markets such as China or Japan. The main transformation power that this technology has is by catalysing the development of autonomous mobility and making it possible to deliver all the benefits for society sooner than expected. As GNSS positioning is a technique that has not played an important role in this transformation yet, the possibilities are immense.

In first place, an accurate geo-positioning system could make it possible for autonomous cars to overcome geofencing as mapping could be unnecessary because the satellite maps that are already available could be used. This would make the car capable of driving and adapt to many other environments quicker than ever.

In second place, another very important thing to take into consideration is the ecological impact of the technology. As the automated driving technology is, in most cases, linked to electric mobility, it is a technology that can catalyse the ecologic transition. Basically, reaching a sufficiently autonomous driving technology would make the number of cars to decrease and even more the pollution as the cars that would be in the streets would be electric. Moreover, the developed technology in this thesis would also have a positive impact in health as, by empowering the cited technologies, it could also reduce acoustic pollution and traffic jams, hence, reducing anxiety and stress in society. Therefore, this technology's success could be beneficial not only for the physical health but also to the mental.

What has been developed in this thesis could also be a key player in the construction of interconnected intelligent information nets between the cars that are in the streets and roads. Another key contribution of intelligent vehicles is the capacity of creating a system in which all cars are tracked in order to optimise traffic with all the social and economic benefits that this has. An accurate positioning technology could provide the most important data to create these systems.

Finally, the purely economic benefits could also be important. An accurate positioning technology could make delivery or transport companies save millions each year because having this high accuracy technique could help optimise the different procedures that the companies go through.

Another application of the prediction models that have been developed in this thesis are navigation systems from a consumer perspective as the ones that exist nowadays in the market are not reliable enough. Situations as the ones that are resolved in this thesis create huge problems to these systems that are unable to overcome.

To conclude, the overall economic viability of scaling the technology is very high because the system has been developed considering the cheapest GNSS systems in the market and the model is proven to be right and accurate. Although it could be explored the possibility of using systems that are more expensive, the value for money of the technology developed in this thesis is absolutely brilliant.

9 CONCLUSIONS AND FUTURE WORK

This thesis is going to conclude with an analysis that will solve the questions that have appeared during the thesis as well as a deep analysis that clarifies the significance of this work itself. In addition to this, some future work proposal will also be made in order to set a path for this investigation to continue in the future.

First of all, based on the obtained results, it is possible to conclude that the performance of the Kalman Filter models is satisfactory. The Constant Velocity Model is just an interesting approach to the matter, but both the Constant Acceleration Model and the Unscented Kalman Filter Model have given promising results. Whereas the CAM has demonstrated to be able to correct errors in GNSS systems, the UKFM has also demonstrated that the error it gives is even lower that in the CAM as the trajectory trace is highly improved in this last model. However, the purpose of this work is to construct a model that can perform corrections in live motion, while a car is driving and, hence, the quickness of the model is very important and the best of them is the one that finds a better equilibrium between accuracy and velocity. For this reason, the best model of all three is the Constant Acceleration Model because is the most equilibrated of all of them. Although the Unscented Kalman Filter Model performs a higher accuracy, it is computationally much heavier because of the mathematics of the model but also because of using an API that makes it possible to use MATLAB functions and files from the Python environment. In conclusion, the proposed model as a solution to the GNSS precision problematic is the Constant Acceleration Model because of its accuracy, quickness and robustness.

The projection of this study is, in any case, very promising. The UKFM still has to be adjusted in order to exploit all its possibilities and, then, it is intended to output a much more accurate model than the CAM. Moreover, the computational load could also be lowered by implementing in Python the UKFM and optimising the mathematical operations. Therefore, there is room enough for the UKFM to improve its quickness.

Also, the model has been developed considering the cheapest sensors in the market that do not have accurate inertial measurement sensors. Therefore, the value for money of the technological solution proposed could be improved by finding a better equilibrium between the cost of a GNSS receptor and the performance of it. In any case, the objective will be to not compromise the scalability of the technology. It is important to note that the actual solution is able to perform in sensors much cheaper that the one that has been used in the thesis and, hence, the scalability of the technology is possible already today.

Finally, this thesis is going to be concluded by drawing the path to be followed in future works that could lead to the development of a localisation technology that could mean a real breakthrough. It is absolutely clear that the base set by this thesis is very promising, and there are some other new technologies that could help create a much better system. One of the possibilities is developing a Simultaneous Localisation and Mapping (SLAM) algorithm that performs a mapping and simultaneous correction technology that could be used for a forward Machine Learning algorithm. A type of Neural Network could be also developed and trained to learn how to correct positioning better and faster and, with that, developing a technology that is capable of perform reliably in any environment and giving

autonomous driving a critical boost. The simplicity of Kalman Filtering makes it easy to create a bigger system from it, because accuracy and quickness are already excellent. In any case, this technology is undeniably promising, and, if followed the path set by this thesis, autonomous driving could become a reality sooner than everyone expects.

10 BIBLIOGRAPHY

[1] *BESTPOS*. (2020). Retrieved 2 October 2019, from https://docs.novatel.com/OEM7/Content/Logs/BESTPOS.htm#SolutionStatus.

[2] *BESTUTM*. (2020). Retrieved 2 October 2019, from https://docs.novatel.com/OEM7/Content/Logs/BESTUTM.htm.

[3] *GPS Reference Time Status*. (2020). Retrieved 2 October 2019, from https://docs.novatel.com/OEM7/Content/Messages/GPS_Reference_Time_Statu.htm.

[4] Lu, D., Jiang, S., Cai, B., Shangguan, W., Liu, X., & Luan, J. (2018). Quantitative analysis of GNSS performance under railway obstruction environment. *2018 IEEE/ION Position, Location And Navigation Symposium (PLANS)*. https://doi.org/10.1109/plans.2018.8373489

[5] Jende, P., Nex, F., Gerke, M., & Vosselman, G. (2018). A fully automatic approach to register mobile mapping and airborne imagery to support the correction of platform trajectories in GNSS-denied urban areas. *ISPRS Journal Of Photogrammetry And Remote Sensing*, *141*, 86-99. https://doi.org/10.1016/j.isprsjprs.2018.04.017

[6] Mur-Artal, R., & Tardos, J. (2017). ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Transactions On Robotics*, *33*(5), 1255-1262. https://doi.org/10.1109/tro.2017.2705103

[7] Yongfang, N. (2018). Scaling parameters selection principle for the scaled unscented Kalman filter. *Journal Of Systems Engineering And Electronics*, *29*(3), 601. https://doi.org/10.21629/jsee.2018.03.17

[8] Schubert, R., Adam, C., Obst, M., Mattern, N., Leonhardt, V., & Wanielik, G. (2011). Empirical evaluation of vehicular models for ego motion estimation. *2011 IEEE Intelligent Vehicles Symposium (IV)*. https://doi.org/10.1109/ivs.2011.5940526

[9] Silver, D. (2017). *All About Kalman Filters*. [online] Medium. Retrieved 15 October 2019, from https://medium.com/self-driving-cars/all-about-kalman-filters-8924abe3aa88.

[10] Srini, S. (2018). *Kalman Filter: An Algorithm for making sense from the insights of various sensors fused together*. Medium. Retrieved 15 October 2019, from

https://towardsdatascience.com/kalman-filter-an-algorithm-for-making-sense-from-the-insights-of-various-sensors-fused-together-ddf67597f35e.

[11] Teow, J. (2018). *Understanding Kalman Filters with Python*. Medium. Retrieved 15 October 2018, from https://medium.com/@jaems33/understanding-kalman-filters-with-python-2310e87b8f48.

[12] Chadha, H. (2018). *Kalman Filter Interview*. Medium. Retrieved 16 October 2019, from https://towardsdatascience.com/kalman-filter-interview-bdc39f3e6cf3.

[13] Li, Hao. A Brief Tutorial On Recursive Estimation With Examples From Intelligent Vehicle Applications (Part III): Handling Nonlinear Estimation Problems And The Unscented Kalman Filter. 2014.hal-01054709

[14] Van der Merwe, R. (2004). *Sigma-Point Kalman filters for Probabilistic Inference in Dynamic State-Space Models* (p. 378).

[15] Chadha, H. (2018). *The Unscented Kalman Filter: Anything EKF can do I can do it better!*. Medium. Retrieved 11 December 2019, from https://towardsdatascience.com/the-unscented-kalman-filter-anything-ekf-can-do-i-can-do-it-better-ce7c773cf88d.

[16] Hartikainen, J., Solin A., Särkkä, S. (2011). *Optimal Filtering with Kalman Filters and Smoothers. A manual for the Matlab toolbox EKF/UKF* (p. 131).

[17] European Parliament (2019). *European Parliament Resolution of 15 January 2019 on autonomous diving in European transport (2018/2089(INI))* (p.13).

[18] Gobierno de España Ministerio del Interior, Dirección General de Tráfico, Subdirección General de gestión de la Movilidad. *Instrucción 157V-113* (p. 49).

[19] Official Journal of the European Union. *Regulation (EU) No 1285/2013 of the European Parliament and of the Council of 11 December 2013 on the implementation and exploitation of European satellite navigation systems and repealing Council Regulation (EC) No 876/2002 and Regulation (EC) No 683/2008 of the European Parliament and of the Council*.

[20] Ministerio de la Presidencia, Gobierno de España. Real Decreto Legislativo8/2004, de 29 de octubre, por el que se aprueba el texto refundido de la Ley sobre responsabilidad civil y seguro en la circulación de vehículos a motor.

11 APPENDIX

Appendix 1: Zoom of the dataset in the .txt. file.

```
1: 24.10.2014.19.2.38.227
#BESTVELA,COM1,0,58.5,FINESTEERING,1815,493260.800,00000000,827b,6988;SOL_COMPUTED,PSRDIFF,0.150,221.000,0.0336,166.242521,-0.0081,0.0*311e9391

2: 24.10.2014.19.2.38.228
#BESTUTMA,COM1,0,58.5,FINESTEERING,1815,493260.800,00000000,eb16,6988;SOL_COMPUTED,PSRDIFF,30,T,4467786.5677,434255.8274,666.3304,52.1000,WGS84,2.9417,2.1741,5.1275,"0",221.000,0.000,17,13,0,0,08,0,01*85624e19

3: 24.10.2014.19.2.38.229
#BESTVELA,COM1,0,58.5,FINESTEERING,1815,493261.000,00000000,827b,6988;SOL_COMPUTED,PSRDIFF,0.150,221.200,0.0266,301.963850,-0.0172,0.0*042c89ef

4: 24.10.2014.19.2.38.230
#BESTUTMA,COM1,0,58.5,FINESTEERING,1815,493261.000,00000000,eb16,6988;SOL_COMPUTED,PSRDIFF,30,T,4467786.5842,434255.8075,666.3376,52.1000,WGS84,2.9461,2.1774,5.1355,"0",221.200,0.000,16,13,0,0,08,0,01*4d7db485

5: 24.10.2014.19.2.38.231
#BESTVELA,COM1,0,58.5,FINESTEERING,1815,493261.200,00000000,827b,6988;SOL_COMPUTED,PSRDIFF,0.150,221.400,0.0289,162.903724,0.0165,0.0*696d6084

6: 24.10.2014.19.2.38.232
#BESTUTMA,COM1,0,58.5,FINESTEERING,1815,493261.200,00000000,eb16,6988;SOL_COMPUTED,PSRDIFF,30,T,4467786.5773,434255.8008,666.3382,52.1000,WGS84,2.9507,2.1809,5.1437,"0",221.400,0.000,16,13,0,0,08,0,01*5c39c3ec

7: 24.10.2014.19.2.38.233
#BESTVELA,COM1,0,58.5,FINESTEERING,1815,493261.400,00000000,827b,6988;SOL_COMPUTED,PSRDIFF,0.150,221.600,0.0509,353.763645,-0.0428,0.0*2e5a22fc

8: 24.10.2014.19.2.38.234
#BESTUTMA,COM1,0,58.5,FINESTEERING,1815,493261.400,00000000,eb16,6988;SOL_COMPUTED,PSRDIFF,30,T,4467786.5758,434255.8083,666.3249,52.1000,WGS84,2.9556,2.1845,5.1520,"0",221.600,0.000,17,13,0,0,08,0,01*55026f3d

9: 24.10.2014.19.2.38.235
#BESTVELA,COM1,0,58.5,FINESTEERING,1815,493261.600,00000000,827b,6988;SOL_COMPUTED,PSRDIFF,0.150,221.800,0.0189,137.506106,-0.0046,0.0*9efc09ed

10: 24.10.2014.19.2.38.235
#BESTUTMA,COM1,0,58.5,FINESTEERING,1815,493261.600,00000000,eb16,6988;SOL_COMPUTED,PSRDIFF,30,T,4467786.5869,434255.7969,666.3262,52.1000,WGS84,2.9600,2.1875,5.1602,"0",221.800,0.000,17,13,0,0,08,0,01*e526426d

11: 24.10.2014.19.2.38.237
#BESTVELA,COM1,0,58.5,FINESTEERING,1815,493261.800,00000000,827b,6988;SOL_COMPUTED,PSRDIFF,0.150,222.000,0.0069,236.559521,0.0064,0.0*b4c123f2

12: 24.10.2014.19.2.38.237
#BESTUTMA,COM1,0,58.5,FINESTEERING,1815,493261.800,00000000,eb16,6988;SOL_COMPUTED,PSRDIFF,30,T,4467786.5816,434255.7895,666.3270,52.1000,WGS84,2.9645,2.1910,5.1682,"0",222.000,0.000,18,13,0,0,08,0,01*5b435093

13: 24.10.2014.19.2.38.238
#BESTVELA,COM1,0,60.0,FINESTEERING,1815,493262.000,00000000,827b,6988;SOL_COMPUTED,PSRDIFF,0.150,222.200,0.0280,238.331365,0.0043,0.0*0d8baaeb

14: 24.10.2014.19.2.38.240
#BESTUTMA,COM1,0,60.0,FINESTEERING,1815,493262.000,00000000,eb16,6988;SOL_COMPUTED,PSRDIFF,30,T,4467786.5717,434255.7904,666.3247,52.1000,WGS84,2.9689,2.1944,5.1763,"0",222.200,0.000,18,13,0,0,08,0,01*df181fcc

15: 24.10.2014.19.2.38.241
#BESTVELA,COM1,0,60.0,FINESTEERING,1815,493262.200,00000000,827b,6988;SOL_COMPUTED,PSRDIFF,0.150,222.400,0.0418,350.771206,-0.0029,0.0*cc44d29c

16: 24.10.2014.19.2.38.241
#BESTUTMA,COM1,0,60.0,FINESTEERING,1815,493262.200,00000000,eb16,6988;SOL_COMPUTED,PSRDIFF,30,T,4467786.5499,434255.8072,666.3246,52.1000,WGS84,2.9735,2.1979,5.1843,"0",222.400,0.000,18,13,0,0,08,0,01*a6cc7358
```

Appendix 2: URL to the GitHub repository in which all the python files are uploaded.

https://github.com/Manu-Fraile/Kalman-Filtering